DOCUMENT RESUME

ED 248 889 IR 011 365

TITLE Courseware Portability. Proceedings of the International Conference of the Association for the Development of Computer-Based Instructional Systems (25th, Columbus, Ohio, May 14-17, 1984).

INSTITUTION Association for the Development of Computer-based Instructional Systems.

PUB DATE May 84

NOTE 281p.; Cover title: "Computer Transportability."

PUB TYPE Collected Works - Conference Proceedings (021) -- Viewpoints (120) -- Reports - General (140)

EDRS PRICE MF01/PC12 Plus Postage.

DESCRIPTORS Business; *Computer Assisted Instruction; *Computers; *Courseware; Government (Administrative Body); Industry; Military Organizations; *Networks; *Training Methods

IDENTIFIERS *Computer Uses in Education

ABSTRACT

This conference collection presents 40 complete papers and abstracts of 33 project reports and 19 poster sessions, representing discussions of a variety of issues in the applications of computers to learning in business, industry, education, government, and the military services. A presentation on the conference topic "Courseware Transportability through Networking," by Greg Kearsley, is followed by papers that are grouped by topics related to ten special interest groups of the Association for the Development of Computer-Based Instructional Systems: (1) computer-based training; (2) elementary/secondary, junior college; (3) educators of the handicapped; (4) health education; (5) home economics consortium; (6) math; (7) mini/micro; (8) music consortium; (9) Plato Users; and (10) theory and research. Most of the complete papers include abstracts and references. Abstracts of project reports describe programs involving interactive video, computer-based instruction, computer-based authoring systems, and other applications of technology to education and training. Poster session abstracts address such topics as interactive computer graphics, an electronic logbook/evaluation system for clinical training, application of hand-held computers in training, simulations of the clinical encounter, interactive videodisc, and software development. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

Proceedings of the 25th International ADCIS Conference

COURSEWARE TRANSPORTABILITY

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED248889

IR011365

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Gordon Hayes
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

2

ADCIS

Association for the Development of Computer-Based Instructional Systems

May 1984

25th ADCIS CONFERENCE PROCEEDINGS

Association for the Development of
Computer-Based Instructional Systems

"COURSEWARE PORTABILITY"
COLUMBUS, OHIO
MAY 14 thru 17, 1984

IR 011365

Copyright© ADCIS 1984
ADCIS, MILLER HALL 409
WESTERN WASHINGTON UNIVERSITY
BELLINGHAM, WA 98225

FOREWORD

The application of computers to learning is gaining broad-based acceptance in business, industry, education, government and the military services. Every day this acceptance grows. With this growth comes a professional responsibility to keep abreast of new developments.

The 25th International Conference is 'the' professional development event of the year. It is not only a singular opportunity to round up information about this year's theme, Courseware Portability, it is also an opportunity to add to your personal and professional data banks. To help you do this, the conference program has been expanded to include a number of sessions dealing with the "how to." Also, by involving the Federal Interagency Group on Computer-Based Training, I am showcasing federal and military achievements in computer-based training.

This year's expanded format testifies to the tremendous growth in our profession. I encourage each of you to look outside your specific job, investigate developments in other areas, learn, and carry those insights back with you. This conference is your opportunity to grow professionally. Embrace it!

Glenn E. Head
Program Chairperson

ADCIS BOARD OF DIRECTORS

President
Ron Comer
The Ohio State University

Vice President-Treasurer
Jan Richardson
Gallaudet

Program Chairperson
Glenn Head
Instructional Communications

JCBI Editor
Allen Avner
University of Illinois

Newsletter Editor
Charles Buchanan
United Airlines

Chapter Council Representative
Rick Anderson
University of Cincinnati

Special Interest Group Chairperson
Bob Tannenbaum
Hunter College

Special Interest Group Representative
Frank Roberts
Control Data Corporation

Special Interest Group Representative
Terris Wolff
University of Southern California

ADCIS Executive Director
Gordon Hayes
ADCIS International Headquarters
Ex-officio

SPECIAL INTEREST GROUP COUNCIL

Computer-Based Training
Robert Fratini
Western Electric

Elementary/Secondary/Jr. College
Mel Samuels
Allegheny Intermediate Unit

Jeffry Gordon
University of Cincinnati

Educators For the Handicapped
Susan Rice
University of Minnesota

Health Education
Arthur D. King
University of Florida

Home Economics
Nancy Dillon
Phoenix, Arizona

Implementation
Terris Wolff
University of Southern California

Interactive Instructional System Presentation and Authoring
Judith Sentieri
Aetna Life & Casualty

Interactive Video/Audio
Donna Larson
Grand Valley State Colleges

Math
Ron Wenger
University of Delaware

Mini/Micro
Bob Tannenbaum
Hunter College

Music
Michael Arenson
University of Delaware

PLATO
Peter Johnston
University of Alberta

Theory & Research
Frank Roberts
Control Data Corporation

REGIONAL CHAPTERS

Front Range
Chuck Buchanan

Pacific Northwest
Ian Rose

Midwest
Chris Dimas

Mideastern
Rick Anderson

TABLE OF CONTENTS

TOPIC PRESENTATION

Courseware Transportability Through Networking 1
Greg Kearsley

COMPUTER-BASED TRAINING (SIG/CBT)

A Computer-Based Career Guidance System for the Army! 7
Melissa S. Berkowitz, Helena F. Barsam

Courseware Transportability Issues Revisited 12
Richard C. Brandt

Courseware Authoring Standards 16
Katie Gurney, Carol Clark

Integrating Computer Graphics and the Videodisc: New Instructional Techniques Possible with IVIS 19
Jesse M. Heines, Lin Olsen, and Roger S. Bowker

Computer-Based Education: Preparation of an Educational Technologist 24
D.L. Jelden

Robotics: Training Challenge of the 1980s? 28
B.K. Lindauer, L.J. Bruemmer

Federal Aviation Administration Computer-Based Instruction — Year 10 34
Barbara Wright, John A. Buck

ELEMENTARY, SECONDARY, JR. COLLEGE (EL/SEC/JC)

Automated Instructional Management - AIM/3000 38
John Ackelson, Ted Rogers

Computer Literacy: Is It Improved by the Use of Microcomputers in Content Courses? 42
Marianne J. D'Onofrio, Jane D. Nate, Mark E. Slama

Reading Comprehension Skills and Computer Literacy 49
Harriet Johnson

Needs Assessment, Instructional Design, and the Use of Microcomputers in Instruction 52
David F. Salisbury

EDUCATORS OF THE HANDICAPPED (SIG/HAN)

Educational Software for Problem Solving by Hearing Impaired Students 58
M. Waldron, Susan Rose, Steven Heinrich

HEALTH EDUCATION (HESIG)

The Evolution, Execution and Evaluation of a "Computer Literacy" Workshop for Health Educators 64
Moon S. Chen Jr., Sheila R. Brill

HOME ECONOMICS CONSORTIUM (SIG/HOMEC)

The Alterlab Lesson: A Simulation Package to Teach the Principles of Pattern Alteration 67
Frances Whitaker Mayhew

SIG/MATH

Teaching Estimation of Measurement Through a Microcomputer Instructional Game 71
George W. Bright

The Electronic Spreadsheet in Mathematics Instruction 75
Samuel W. Spero

MINI-MICRO

An Example of Authoring Guidelines Used in Developing Microcomputer-Based Training Courses 82
James A. Cook

Does Learning to Program Improve Children's Thinking Skills? 88
Erik De Corte

The Conversion of PLATO® Courseware to the Apple® Microcomputer 92
Louisa Frank, Lynn Smith

Evaluating Classroom Microcomputer Courseware 97
Paul Geisert, Mynga Futrell

Automated Holistic Scoring of Essays 104
Sue Legg, Vernon Miles
Interfacing Your Micro-Computer to the Real World 108
Robert S. Tannenbaum, David C. Tannenbaum
DELPHI: A Computer Program Featuring Dialogue Among Students and Between Students and the Instructor 117
Jerome H. Woolpy
Development of a Base for Computer-Assisted Learning Methods 123
Hanan Yaniv

MUSIC CONSORTIUM (NCCBMI)

Educational Music Systems for Microcomputers 129
Susan Jenkins
Computer Presentation of Acoustic Spectrum Patterns as an Aid in Teaching Music Performance Skills 133
Peter Simpson, Allen Goodwin

PLATO USERS (PUG)

A Plato-Based English for Science Project 138
Jannie J. Botha
Developing a CBI Course: The Process 142
Roland Garton, Mary Jac Reed, George Reed, Evelyn Stevens
Programming a CBI Course: A Case Study 149
Roland Garton, Mary Jac Reed, George Reed, Evelyn Stevens
A Model for the PLATO® Services Organization in a Large, Multi-System CBE Environment 156
Rae D. Stabosz, Jessica Weissman
Learning About Supercomputers on a Microcomputer With No Keyboard: A Science Museum Exhibit 163
Mary L. Stoddard, Bill L. Buzbee

THEORY AND RESEARCH (SIGTAR)

The coming of Age of Educational Technology: Meta-Analyses of Outcome Studies of Computer-Based Instruction 163
Robert L. Bangert-Drowns, James A. Kulik, Chen-Lin C. Kulik
Using Interactive Logo Procedures to Enhance Clinical Interviews with Children 172
Larry Flick
Skill Theory: The Process for Finding the Lost Content 179
Catharine C. Knight, Walter J. Kuleck
Recall of Verbal Spatial Information in Graphic Form 185
Marie Knowlton
Instructional Graphics in Commercial Courseware 188
Kathryn Lutz Alesandrini
Early Childhood Educational Majors 196
Jack V. Powell, L. David Weller
Strategies for Designing Drill and Practice Programs for Computers 201
David F. Salisbury
The Need for Adaptive Control of Aids and Feedback in Problem-Solving Instructions 208
Esther R. Steinberg, A.B. Baskin, Terri Darling Matthews

PROJECT REPORTS

Library Use Instruction: The Card Catalog and Wilson Indexes 214
Rob Aken, Laura Olson
NAIT's Alternate Training Delivery Project 215
Mohamed Ally
Center for Computer Literacy 216
Ruth Berggren
Interaction of Feedback and Anxiety During on-Line Testing 217
Gerald J. Blumenfeld, Lea Banner, R.O. Blackwood
The National Demonstration Center for Educational Technology 218
Victor H. Carr Jr., Michael Weisberg

Developing and Implementing a Computer-Assisted Instructional Module for Diagnostic Prescriptive Teaching 219
Bert Chiang
Cognitive Abilities and CAI Performance Among Mentally Retarded Adolescents 220
Frances A. Conners
Computer Literacy "A Must in Today's Society" 221
Patricia Veasey D'Souza
Feasibility Analysis Procedures and Instructional Design Plan for Interactive Video Instructional Materials Development 222
Allen DeGraw, Hyoja Lee
Effect of Microcomputer Simulations on Computer Awareness and Perception of Environmental Relationships Among College Students 223
Rosanne W. Fortner, John F. Schar
Experience With a Sophisticated Computer-Based Authoring System 225
Patrick R. Gardner
WITSCHEM 226
L. Glasser, J.D. Bradley, G. Brink
Model Training Program for Reserve Component Units 227
Stephen L. Goldberg
Development of Computer-Based Instruction at the Army Engineer School 228
Wayne D. Gray, Christopher P. Werle
Introduction to the Fundamentals of Music 229
Harriet I. Hair
Reported Availability of Microcomputers in American Schools: Year 2 230
Gary M. Ingersoll, Carl B. Smith
Microcomputers and Mathematics Curriculum Constructing 231
Jim Kelly
Helping Medical Faculty and Students Overcome Computerphobia 232
Lloyd A. Lewis
A Model for Computer-Based Problem Solving in Apparel Design and Pattern Making 234
Ruth Marshall
Computer Testing in an Internal Medicine Clerkship 235
Mary Medal, Lynn Kerbeshian, Reed T. Keller
Managing an International Program by Computer Communications 236
Karen Miller
Computer Technology Program 237
Jim Pollard
Mathematics Enrichment Program 238
Thomas Ralley
Development of Intelligent Videodisc Training Materials for Science Teachers 239
Thomas C. Reeves, James Okey, James M. King
Boston Compact CAI/CMI Initiative 240
Mary Russo
Application of Aesthetic Learning Models for the Appreciation of Computer Art 241
Penelope Semrau
Computer-Assisted Instruction in Laboratory Science - The George Washington University Medical Center 243
Sylvia Silver
The CERL Cluster System 244
Bruce Arne Sherwood
Experimental Psychology Pedagogy: Obesity Simulation 245
John Slaughter, Thomas Rywick
NCR Pilot 246
Timothy W. Spannaus, Pat O'Brien
Computer-Assisted Video-tape Learning Modules 247
Louise Timmer-Hawck
Extension Home Economics: Learn How to Design CAI by CAI 248
Emily S. Wiggins
Surgical Specialty Curriculum Development at the Mount Sinai School of Medicine 249
David P. Yens, Joan T. Peters

POSTER SESSIONS

Interactive Computer Graphics and the Transition from Concrete Manipulation of Objects to Abstract Thinking in Elementary School Mathematics 250
Donna Berlin, Art White

An Electronic Logbook/Evaluation System for Clinical Training 251
L. Brenner, J. Cowan. T. Wacaser

Application of Hand-Held Computers in Training 253
Larry Francis, Gerard Chevalaz, John Merrill

Chemical Formula Entry and Display in WITSCHEM 254
L. Glasser, J.D. Bradley, G. Brink, S. Langley

Introducing Picture Clauses: A Tutorial Program for COBAL Students 255
Karen H. Guthrie

Teacher In-Basket: Stuart and Juanita 256
S. Langham-Johnson

Ecological CAI 112 257
S. Langham-Johnson

Training Teachers to Use Microcomputers 258
Helen Lee

Computer-Aided Teacher Evaluation: Flanders' Interaction Analysis on the Apple 259
JoAnne Martin-Reynolds, Charlotte Scherer

Corporations, Computers, Community and Summer Tech '83 261
Robert E. McNemar

CAI Study Program for the General Psychology College Course 262
Terry F. Pettijohn

Monitored Simulations of the Clinical Encounter 263
Jane Santry-Pierce, Michael Altman, Suzanne Bassani, Barbara Skillman

A CAI System for Sight-Singing 264
Martin Prevel

S T A R S (Space Time Army Reconnaissance System) 265
Thomas C. Reeves, William D. Aggen

Using Microcomputers to Teach Introductory Statiscs 266
Sam Sherrill

Author Transportability — Advantages and Disadvantages 267
Sylvia Silver, Tracie Jenkins

Interactive Videodisc 268
Timothy W. Spannaus

Developing Support Materials for Greater Learner Control of Computer-Based Instruction 269
Brent G. Wilson

Software Development for the Home Ec Classroom 270
Clo Wiltse

PROCEEDINGS INDEX

A

ACKELSON, JOHN 38
AGGEN, WM. D. 265
AKEN, ROB 214
ALLY, MOHAMMED 215
ALTMAN, MICHAEL 263

B

BANGERT-DROWNS, ROBERT 168
BANNER, LEA 217
BARSAM, HELENA 7
BASKIN, A.B. 208
BASSANI, SUZANNE 263
BERGGREN, RUTH 216
BERKOWITZ, MELISSA 7
BERLIN, DONNA 250
BLACKWOOD, R.O. 217
BLUMENFELD, GERALD J. 217
BOTHA, JANNIE 138
BOWKER, ROGER S. 19
BRADLEY, J.D. 226-254
BRANDT, RICHARD C. 12
BRENNER, LISA 251
BRIGHT, GEORGE W. 71
BRILL, SHEILA 64
BRINK, G. 226-254
BRUEMMER, LAWRENCE J. 28
BUCK, JOHN 34
BUZBEE, BILL L. 163

C

CARR, VICTOR H. 218
CHEN, MOON S. 64
CHEVALAZ, GERARD 253
CHIANG, BERTTRAM 219
CLARK, CAROL 16
CONNORS, FRANCES A. 220
COOK, JAMES A. 82
COWAN, J. 251

D

DARLING MATTHEWS, TERRI 208
DeCORTE, ERIK 88
DeGRAW, ALLEN 222
D'ONOFRIO, MARIANNE J. 42
D'SOUZA, PATRICIA VEASEY 221

F

FLICK, LARRY 172
FORTNER, ROSANNE 223
FRANCES, LARRY 253
FRANK, LOUISA 92
FUTRELL, MYNGA 97

G

GARDNER, PATRICK 225
GARTON, ROLAND 142-149
GEISERT, PAUL 97
GLASSER, L. 226-254
GOLDBERG, STEPHEN L. 227
GOODWINN, ALLEN 133
GRAY, WAYNE 228
GURNEY, KATIE 16
GUTHRIE, KAREN H. 255

H

HAIR, HARRIET 229
HEINES, JESSE 19
HEINRICH, STEVEN 58

I

INGERSOLL, GARY M. 230

J

JELDEN, D.L. 24
JENKINS, SUSAN 129
JENKINS, TRACIE 267
JOHNSON, HARRIET 49

K

KEARSLEY, GREG 1
KELLER, REED T. 235
KELLY, JIM 231
KERBESHIAN, LYNN 235
KING, JAMES M. 239
KNIGHT, CATHARINE 179
KNOWLTON, MARIE 185
KULECK, WALTER J. 179
KULIK, CHEN-LIN 168
KULIK, JAMES A. 168

L

LANGLEY, S. 254
LANGMAN-JOHNSON, S. 256-257
LEE, HELEN 258
LEE, HYOJA 222
LEGG, SUE 104
LEWIS, LLOYD A. 232
LINDAUER, BARBARA K. 28
LUTZ-ALESANDRINI, KATHRYN 188

M

MARSHALL, RUTH 234
MARTIN-REYNOLDS, JO ANNE 259
MAYHEW, FRANCES 67
McNEMAR, ROBERT E. 261
MEDAL, MARY 235
MERRILL, JOHN 253
MILES, VERNON 104
MILLER, KAREN 236

N

NATE, JANE D. 42

O

O'BRIEN, PAT 246
OKEY, JAMES 239
OLSEN, LIN 19
OLSON, LAURA 214

P

PETERS, JOAN T. 249
PETTIJOHN, TERRY F. 262
POLLARD, JIM 237
POWELL, JACK V. 196
PREVEL, MARTIN 264

R

RALLEY, THOMAS 238
REED, GEORGE A. 142-149
REED, MARY JAC 142-149
REEVES, THOMAS C. 239-265
ROGERS, TED 38
ROSE, SUSAN 58
RUSSO, MARY 240
RYWICK, T HOMAS 245

S

SALISBURY, DAVID F. 52-201
SANTRY-PIERCE, JANE 263
SCHAR, JOHN F. 223
SCHERER, CHARLOTTE 259
SEMRAU, PENELOPE 241
SHERRILL, SAM 266
SHERWOOD, BRUCE ARNE 244
SILVER, SYLVIA 243-267
SIMPSON, PETER 133
SKILLMAN, BARBARA 263
SLAMA, MARK E. 42
SLAUGHTER, JOHN 245
SMITH, CARL B. 230
SMITH, LYNN 92
SPANNAUS, TIMOTHY 246-268
SPERO, SAMUEL W. 75
STABOSZ, RAE D. 156
STEINBERG, ESTHER R. 208
STEVENS, EVELYN 142-149
STODDARD, MARY L. 163

T

TANNENBAUM, DAVID 108
TANNENBAUM, ROBERT 108
TIMMER-HAWCK, LOUISE 247

W

WACASER, T. 251
WALDRON, M. 58
WEISBERG, MICHAEL 218
WEISSMAN, JESSICA 156
WELLER, DAVID L. 196
WERLE, CHRISTOPHER P. 228
WHITE, ART 250
WIGGINS, EMILY S. 248
WILSON, BRENT G. 269
WILTSE, CLO 270
WOOLPY, JEROME H. 117
WRIGHT, BARBARA 34

Y

YANIV, HANAN 123
YENS, DAVID P. 249

Courseware Transportability Through Networking

Greg Kearsley

Courseware, Inc.

ABSTRACT

This paper discusses the use of communications networks as a potential solution to the transportability of CBI courseware. Historically, remote or local CBI networks have been used only to provide access to a specific system (e.g., PLATO, TICCIT, IIS). However, with the emergence of public networks, protocol converters, and gateways which interconnect different networks, it is now possible for the same terminal to access numerous CBI systems. Current issues and problems in the use of networks for CBI include terminal incompatabilities, lack of appropriate networking software, and lack of experience with online group instruction. The impact of new technological developments on CBI networking such as videotex, handhelds, and videoconferencing are briefly considered.

The lack of transportability of courseware represents one of the major obstances to the widespread adoption and use of Computer Based Instruction (CBI). Organizations and commercial vendors are reluctant to invest the significant resources required to develop high quality courseware, if these materials are limisted to a specific delivery system. One of the reasons why overheads, slides, audiocassettes, and videotapes have become established as instructional media is that they use standardized delivery equipment. Absence of standardization for CBI systems and courseware prevents its similar acceptance in the education and training domain.

There are basically three potential solutions to the transportability problem. The first involves the use of a common operating system across different CBI systems. In the mainframe world, the use of IBM operating systems such as OS/VS represents a kind of de facto standard (but only for IBM compatible equipment). In the microcomputer world, CP/M and MS-DOS have become de facto standards. The trouble with this approach, however, is that while these systems may be used by many, they are not used by everyone. For example, there are LSI/11 and Unix systems. Furthermore, operating systems change and new versions are often not completely compatible with earlier ones.

The second solution to the transportability problem is the use of a highly portable code which can easily be converted to different systems. Many organizations are developing courseware in languages such as BASIC, PASCAL, or C and implementing it on multible brands of personal computers. For example, CDC has developed some of its PLATO educational courseware to run on TI, Apple, and Atari microcomputers and many publishers are doing the same thing. Some authoring languages such as PILOT and PLANIT exist on a variety of systems and the code is relatively easy to transfer. Again, this is not an entirely satisfactory solution because of the costs involved with transfer to each different machine and the fact that particular models of microcomputers sometimes have a relatively short lifetime.

A third solution to the problem of courseware transportability is communication networks. When courseware is stored in a central database, any kind of terminal which can be connected to that database can share the courseware. This,

of course, was the idea behind the design of timesharing systems and their current offspring, distributed systems. However, the early timesharing systems often used unique communications or display protocols which made them accessible only to certain kinds of terminals. This was certainly true of the original PLATO, TICCIT, IIS, AID, and other major CBI systems.

Today, some of these systems can use a standard ASCII asynchronous protocol and can communicate with different types of terminals. For example, IBM personal computers can link into the PLATO network via MicroLink and use courseware which does not require touch input. TICCIT and IIAS/PS can also use IBM personal computers as terminals. AID-II can use a variety of different types of terminals. This capability of using a standard microcomputer as a CBI terminal overcomes the transportability problem. However, there are still incompatability and other issues associated with this approach.

The rest of this paper discusses the detailed considerations associated with the use of communications to achieve courseware transportability. The two other approaches to transportability discussed above are viable alternatives and may ultimately be the best solutions. However, emerging telecommunications technologies such as videotex, videoconferencing, direct broadcast satellites, and cellular radio, along with the development of highly portable and powerful handheld computers, suggests that communications will play a major role in future CBI activities. If this turns out to be the case, then working on transportability through networking may be a prudent strategy.

Network Alternatives

There are a number of different network alternatives available which could be used for CBI systems. The two major types of network currently in use are remote and local networks. Remote networks involve the use of telecommunications between a central computer and database, and the terminals or microcomputers used for instructional delivery or development. Local networks do not involve telecommunication links and usually connect terminals/microcomputers in the same building or campus by means of cabling.

Remote systems use either asynchronous or synchronous communications protocols. ASCII is a widely used asynchronous protocol which is normally transmitted at speeds of 300 or 1200 baud. The most common synchronous protocols are bisynchronous and SNA/SDLC which involve the faster transmission speeds of 2400 - 9600 baud (the higher speeds requiring dedicated digital data lines). Synchronous protocols tend to be used only by IBM compatible systems whereas asynchronous protocols (e.g., ASCII) are used by everybody.

There are a number of different local network architectures and transmission protocols. The most common are terminals and microcomputers cabled directly to mainframe controllers by means of coax cable and using high speed synchronous protocols. Local networks based upon the Arcnet or Ethernet architectures have also come into use. Many vendors make local network interfaces for popular microcomputers (e.g., Corvus, Nestar, Davong, Orchard, Tecmar, etc.). In fact, some of these local networks allow different brands of microcomputers to be hooked together in the same network.

While individual microcomputers in a local network can often communicate directly with each other, they normally communicate through a network server. The network server is the central processor in the network which routes all messages and contains all shared databases. Network servers are usually personal computers equipped with a large amount of memory (e.g., 256K) and Winchester disks with capacities of 6-100 megabytes of storage.

Remote or local networks can be either timeshared or distributed systems. In a timeshared system, all processing is done on the central computer (even though it might be another microcomputer) and the terminals/microcomputers simply act as input/output devices. In a distributed system, processing is done locally by the terminals/microcomputers and data is downloaded/uploaded between the central processor and the terminals. The role of the central processor in a distributed network is to manage the network, not to do computation for network terminals.

Let's relate these different alternatives to CBI systems. In the past, all CBI systems were classic timesharing systems with no local processing capability. PLATO was the epitome of a large scale remote system designed to connect hundreds of terminals all over the country via telephone lines. With the development of the MicroPLATO terminal (CDC 110), PLATO became a distributed system capable of downloading portions of courses and also running local courseware. IBM IIS (now IIAS/PS) ran on IBM compatible mainframes and could be used in a local or remote network that used synchronous protocols and 3270 type terminals. Now it is possible to connect

IBM personal computers equipped with appropriate interfaces to IIAS/PS local or remote networks. The new PCIS authoring system permits authoring on the IBM personal computer and uploading/ downloading courseware with a mainframe system. TICCIT was originally a local network using special purpose terminals. Now Hazeltine MicroTICCIT can connect remotely to suitably equipped IBM personal computers using an ASCII protocol.

Users are no longer tied to specially designed systems for CBI, however. Almost any recent microcomputer has all the capabilities needed to deliver CBI courseware including color graphics, sound, and non-keyboard input modes such as touch, lightpen, joystick, or mouse. Authoring languages and systems (or general purpose languages and software tools) can be used to develop courseware on these machines. The capability of connecting different brands of microcomputers in remote and local networks is easily available, making sharing of courseware possible independent of any CBI system.

Communication Components

To equip a microcomputer for communications with a remote network, three basic components are needed: a serial interface, a modem, and communications software. The serial interface is contained inside the computer and connects to the modem by means of an RS-232C connector. Modems come in a variety of types, but the most common today is the direct connect 300/1200 baud asynchronous modem. This type of modem connects directly with the phone outlet by means of a module jack (i.e., no telephone is needed). Some manufacturers now make serial interfaces with the modem included so that you simply plug the modular phone cord from the computer into a telephone outlet. Manufacturers also make interfaces that include both asynchronous and synchronous communications capability on the same card so that your computer can use either protocol.

Communications software provides the microcomputer with the capability of transfering files and storing them on diskette. Thus, it is the communications software which permits the uploading/ downloading capability of distributed systems. This software also allows you to change the parameters of communications protocols and switch from batch to interactive transmission. It also allows you to set display parameters such as width, page size, display speed, etc. There are many different brands of communications software on the market featuring different levels of sophistication and usability. You do not have to use the same brands or machines in order for them to be able to communicate with each other or a network (although it helps).

To use a microcomputer in a local network, you need an interface card and communications software. The interface and software will be unique to the particular type and brand of local network. Each microcomputer in the network must have the interface installed and be connected (via cable, twisted wire, or optical fiber) to the server or other terminals. To be connected to the local network, the terminal must be within a certain distance ranging between 100 feet and 4 miles depending upon the type and brand of the network.

Remote and local networks can be inter-connected. To connect two local networks, a telecommunications link is established between the two servers (or any two terminals) in the network. These two links are called the gateway nodes. The link may be directly between the two local networks or it may pass through a mainframe host. In addition, a local network can be linked to a mainframe, either directly or via telecommunications. Since these gateway nodes may involve different protocols (e.g., asynchronous to synchronous), the machines which serve as nodes often have hardware or software protocol converters.

Networks can be public or private. Most organizations have established their own private networks for administrative or billing functions. The travel and hotel industry has developed private networks for reservation information. Banks and financial institutions have networks for remote financial transactions and credit information. These private networks are sometimes used for CBI as in the case of IIS type systems which piggyback on existing hardware and software. CDC PLATO represents a public CBI network to which anyone can purchase access. Hazeltine TICCIT represents a private CBI local network (with remote capability) which can only be used in the locale where it is installed.

Finally, mention should be made of public networks such as the Source, CompuServe, or EIES. These information utility networks are available to anyone who wants to buy an account and pay for usage. Such systems are the forerunners in North America of videotex networks already in place in Europe (see discussion below). Information utilities rely on the packet switched common carrier networks

(such as Telenet, Tymnet or Uninet) to achieve local telephone nodes across the country. Any individual or company can use these information utilities for CBI purposes. In fact, a small sample of educational courseware is available on the Source and CompuServe.

A company called Telearning Systems is offering dozens of online courses which involve computer conferencing between instructors and students. The New Jersey Institute of Technology EIES system also offers electronics seminars on about 20 different topics. Electronic seminars and information utilities represent relatively unexplored areas for CBI.

Issues and Problems

From the foregoing, it can be concluded that a wide range of alternatives for sharing CBI courses are afforded by communications networks. Some of these alternatives exist within the context of specialized CBI systems such as PLATO, IIS or TICCIT. Others simply involve the transfer of courseware between standard microcomputers equipped with some kind of communications capability. Technically, the transmission or transfer of courseware or student performance data are no different than accounting records or reservation information.

However, there are special properties of CBI courseware which make it different from other data/programs and hence pose some problems for networking. A lot of CBI courseware involves graphics and color which cannot be displayed on certain microcomputers with monochrome, alphanumeric type monitors. Courseware which takes advantage of non-keyboard input modes often cannot be used by a variety of different machines (unless the use of cursor keys is programmed as a standard alternative). Similarly, sound/speed output does not travel well across different machines (although it can be transmitted perfectly well since it is digitally encoded).

What this boils down to is that the closer the courseware comes to being plain vanilla type CBI (i.e., alphanumeric), the greater its chances of complete transportability via communications. This is where videotex could have substantial impact on CBI. If videotext systems come into widespread us in the U.S., large numbers of terminals (either TVs with decoders or microcomputers with videotex interfaces) capable of displaying the same kind of color graphics courseware would exist. Of course, this possibility is clouded by the fact that there are currently three or four different (and incompatible) videotex protocols being tried out in the U.S.

A significant limitation on the use of communications networks for CBI at present is the lack of suitable networking software. As mentioned earlier, software is easily available which performs the necessary transmission functions. However, this software is not specifically designed for instructional applications and does not make it especially easy to accomplish CBI functions. For example, in a distributed CBI environment, the student would probably have to initiate the transfer of student performance data back to the central database (either remote or local) after finishing a lesson or course. Chances are that the student would have to use operating system level commands which would not be especially friendly.

Furthermore, it is unlikely that software to merge the transmitted student data with the rest of the student records would be commercially available. It would have to be written for this purpose. The point is that the capabilities (both hardware and software) exist to do this essential CBI activity but not in a particularly usable form. At the present time, those who attempt to make extensive use of communication networks will be involved in "do-it-yourself" CBI.

There are currently two ways to avoid this problem. One is to stick to a commercially available CBI system in which the necessary software has been written and comes with the system. The limitation of this approach is that you must live with the particular communications capabilities provided. Often this approach also means buying that vendor's CBI terminals/microcomputers (even though you may plenty of your own already). The second way of avoiding the problem is to make use of an information utility. While these systems will not have all the special purpose CBI software required, they are designed to be highly usable and reliable for interactive computing. Furthermore, they are relatively inexpensive ways of doing CBI since you need only pay for initial subscriptions and actual usage.

The discussions of information utilities brings up a more fundamental issue in the use of networks for CBI. To date, almost all our attention in the field of CBI has focused on the delivery of individualized instruction. However, a lot of learning involves group instruction and activities. This includes group problem solving and decision making with shared access to common databases. Only in military crew training simulators, has

group CBI been attempted. Yet there are a great many educational and job training situations where group interaction skills are important. The emergence of electronic seminars provides an opportunity to study group CBI further.

Our conception of CBI has tended to stick with the model of courseware as tutorials, drills, simulations, or case studies. However, an equally important aspect of CBI today is teaching people how to work with and use computers as tools (i.e., computer literacy). This includes word processing, spreadsheets, database management, decision support systems, and the specialized software used in job activities. In many cases, access to the programs or databases needed will require a communications link. So, in some CBI circumstances, communications will be necessary to conduct the kind of instruction involved. This means that CBI specialists will need to become as familiar with communications as any other instructional media or approach.

Emerging technologies are likely to make communications an even more critical component to CBI. The potential significance of videotex has already been discussed. Videoconferencing is already well accepted and used in large corporations for certain applications such as product announcement and sales meetings. However, it is simply too expensive at present to use for routine training or educational functions. In the near future, inexpensive videoconferencing equipment may become available based upon microcomputers and direct broadcast satellite technology. Once the equipment becomes easily available, there is no doubt that it will be used widely for instructional applications. This will add a new twist to CBI networking since real-time interaction between instructors and students can be mixed with prepared interactive video sequences.

The increasing use of handheld microcomputers is also going to have substantial impact on networking and CBI. Once suitably inexpensive handhelds with sufficient memory and full graphics displays are widely available, there will likely be great demand for network services of all types. People will want to access remote databases, communicate with others, and tie into local office networks from their personal handhelds. Communications are likely to be via cellular radio or direct broadcast satellite transmissions rather than telephone connections.

Instruction will be one of the important uses for these "electronic books." Development of courseware for handhelds will probably become as diverse an undertaking as the writing and publishing of books is today. In fact, the same authors and organizations may be involved since this could take place within the decade. The same design and authoring techniques we already use for CBI courseware on desktop microcomputers will apply.

Conclusions

The use of communication networks to achieve courseware transportability has been discussed. Commercially available CBI systems such as PLATO, TICCIT and IIAS/PS have been moving towards more flexible, distributed networks. In addition, existing hardware and software allows standard microcomputers to connect with remote or local networks and hence share courseware. The major problems encountered in using networks to share courseware are terminal incompatability, inadequate networking software, and failure to exploit the potential for group instruction and interaction.

There is no way to tell at present whether communications is a good solution to overcoming courseware transportability obstacles relative to the other solutions mentioned in the introduction. Actually, many of the issues associated with the transportability problem are independent of any of these particular solutions. Courseware that is modularly designed and coded in a structured fashion, properly documented both offline and online, and is easy to use, will likely be highly transportable no matter what approach is used. And conversely, that which isn't, won't be. There will always be the "not-invented-here" syndrome to deal with; some (most?) instructors will simply not use courseware they have not created themselves.

Perhaps the ultimate solution to courseware transportability is quality software. The microcomputer market has shown that outstanding software such as Visicalc or Pac-Man will get widely disseminated on different machines because of popular demand. When we develop interactive instruction of such compelling magnitude, transportability issues may vanish. The courseware will get replicated (somehow) due to competitive forces. Of course, this line of reasoning suggests that a major obstacle to courseware transportability lies in the poor quality of the programs we have developed to date. This is a much harder problem to solve than the lack of transportability of courseware.

Bibliography

Davis G. The Local Network Handbook. N.Y.: McGraw-Hill, 1983.

Derfler, F.J. & Stallings, W. A Managers Guide to Local Networks. Englewood Cliffs, N.J.: Prentice-Hall, 1983.

Glossbrenner, A. The Complete Handbook of Personal Computer Communications. N.Y.: St. Martins Press, 1983.

Martin, J. Future Developments in Telecommunications. Englewood Cliffs, N.J.: Prentice-Hall, 1977.

Martin, J. The Telematic Society. Englewood Cliffs, N.J.: Prentice-Hall, 1981.

Nilles, J. Micros and Modems. Reston, VA.: Reston Pub., 1983.

Shapiro, N. The Small Computer Connection: Telecommunications for the Home and Office. N.Y.: Micro/McGraw-Hill, 1983.

Sigel, E. The Future of Videotex. White Plains, N.Y.: Knowledge Industry Publications, 1982.

Turoff, M. & Starr, R.H. The Network Nation: Human Communication via Computer. Reading, MA.: Addison-Wesley, 1978.

Tanenbaum, A. Computer Networks. Englewood Cliffs, N.J.: Prentice-Hall, 1981.

Tydeman, J. et al. Teletext and Videotex in the United States. N.Y.: McGraw-Hill, 1983.

Williams, F. The Communications Revolution. Beverly Hills: Sage Publications, 1982.

A COMPUTER-BASED CAREER GUIDANCE SYSTEM FOR THE ARMY[1]

MELISSA S. BERKOWITZ—US ARMY RESEARCH INSTITITE

HELENA F. BARSAM—SCIENCE APPLICATIONS, INC.

ABSTRACT

This paper will present the rationale, design, and evaluation plan of a microcomputer-based educational and vocational information system being developed for the Army. The Army Education Information System (AREIS) concept evolved from a needs assessment conducted at Army Education Centers (AEC) worldwide. AREIS has been designed to support AEC counselors by providing soldiers with educational and vocational information about military and civilian occupations. AREIS provides soldiers with the opportunity to assess their work-related interests, aptitudes, and values. Soldiers interact directly with the system to identify their short and long-range military, vocational, and educational goals. ARIES has been designed to reduce the administrative workload of counselors, enabling them to spend more time on one-to-one counseling. The system is being evaluated for reliability and user acceptance under operating conditions at three Army posts.

One objective of the US Army is to produce a combat-ready force through the development of personal skills and military proficiency. The Army Continuing Education System (ACES) supports this objective by providing educational opportunities to soldiers and enabling them to develop career goals that include military service and post-service education and training. Army Education Centers (AEC) have been established at every Army post having a minimum of 750 military members. These centers provide programs which are designed to (1) satisfy the skill development and occupational needs of the Army, 2) increase soldier potential, 3) enhance job satisfaction, and 4) increase personal educational growth. Specific offerings for academic education include the Basic Skills Education Program (BSEP) high school completion program, and Service-members Opportunity Colleges Associate Degree (SOCAD) program. In skill development each AEC procides language, military occupational specialty (MOS), and occupation-oriented courses, In the area of skill recognition the Army Apprenticeship Program and Defense Activity for Non-Traditional Education Support (DANTES) certificate training are offered. Each AEC also provides education services which include counseling , testing, and the support of a learning center.

The Adjutant General's Office (TAGO) supervises ACES and develops policy and guidance. Each installation/ community commander conducts the ACES program through the Education Services Officer (ESO) who insures that the ACES educational and vocational opportunities are made available to all service members. The ESO supervises these programs provided by contract instructors and Education Center counselors. The counselors are required to provide each service member with program information and counseling during initial training. Counseling also must be provided within 30 days of arrival at new duty stations, annually during the first enlistment, and 30 days prior to separation. The counseling emphasizes military professional development and educational benefits.

The primary means of delivering information about educational and vocational opportunities rests with the Education Center counselors. Two developments have hampered the provision of services by the Education Center staff: the increasing quantity and complexity of educational and vocational options and the reduction in the number of Education Center counselors. Education Centers, increasingly understaffed for the increased workload, are experiencing difficulty in adequately serving their clients.

It has become evident that other means of supplying standardized, up-to-date, easily accessible educational and vocational information are needed. One such means is a computer-based information system. Over the past two decades, a growing number of guidance professionals have become increasingly committed to the use of the computer to assist with access and delivery of individualized educational and vocational information (Katz & Shatkin, 1980). The unique capabilities of the computer to store, search, retrieve, and update large masses of information; to relate educational and vocational data to information about the user; to simulate an interactive dialogue; and to serve many users simutaneously with tailored information have validated the worth of this technological aid to the counseling process.

A computer-based career guidance system is intended to function in concert with, not instead of, the activities performed by guidance counselors. As the computer carries out information retrieving and dispensing functions and clerical duties, counselors should gain time to perform the professional duties for which they were trained and for which they are needed--one-to-one interviewing, group guidance, and consultation.

This paper will present the design, development, and evaluation plan of a prototype computer-based educational and vocational information system which is one effort to overcome the increase of guidance information and decrease of counseling personnel in the Army.

THE ARMY EDUCATION INFORMATION SYSTEM

Needs Assessment. The US Army Research Institute for the Behavioral and Social Sciences (ARI) initiated a research effort in 1979 to conceptualize and develop a prototype computer-based system which would provide information on military and civilian education programs related to the Army career progression. This effort, performed by DISCOVER/ACT, under contract MDA 903-79-C-0279, provided a design for the Army Education Information System (AREIS) based on the results of a needs assessment survey administered to Education Services Officers (ESOs) and Education Center counselors at Army posts worldwide(Harris-Bowlsbey&Rabush,

1979).

The needs assessment instruments were designed to collect data concerning 1) demographic information about Education Center, 2) the variety and frequency of information requested by soldiers at the Education Center, and 3) ESO and counselor attitudes about using computers. The instruments were distributed to all major commands. The return rate for the ESOs was 72% with 131 of 183 questionnaires mailed back. The return rate for counselors was 64% with 313 of 494 counselors responding.

The following summarizes the demographic data supplied by the ESOs. Of 144 posts responding, the permanent population ranged from 50 to 48,000 as noted in Table 1.

Table 1
Number of Soldiers on Post By Location

	CONUS		
	TRADOC	FORSCOM	OTHER*
Minimum	300	400	200
Maximum	25,000	48,000	18,000

	USAREUR	FAR EAST
Minimum	50	200
Maximum	35,612	8,500

*Other refers to: DARCOM, US Military District, MTMC, INSCOM, WESCOM, US Health Services Command, US Army Communication Command, and US Army Military Academy.

ESOs reported the number of Education Center counselors on their staffs as ranging from 2 to 12 (see Table 2).

Table 2
Mean Number of Counselors on Post By Location:

CONUS				
TRADOC	FORSCOM	OTHER	USAREUR	FAR EAST
5.5	11.5	3.3	4.0	2.2

Counselor/Soldier Ratios were reported by ESOs and counselors as indicated in Table 3. Data indicated that each counselor serves between 1,000 and 2,000 soldiers annually. The average workload is 1,600 soldiers per counselor per year.

Table 3
Mean Counselor/Soldier Ratio By Location

	CONUS		
	TRADOC	FORSCOM	OTHER
ESO-reported	1,835	1,564	992
Counselor-reported	1,567	1,952	1,209

	USAREUR	FAR EAST
ESO-reported	1,359	1,216
Counselor-reported	1,458	2,019

Counselors indicated that half of their time is spent on one-to-one counseling of soldiers with the remainder distributed over administrative duties, orientation/outreach program, clerical duties, liaison efforts, research and development, and other miscellaneous tasks. Counselors provide an average of two interviews per soldier per year. This represents approximately 64% of their workload. ESOs and counselors ranked tuition assistance, college course offerings on or near post, and information about tests as ACES program information most frequently requested by soldiers. In descending order of frequency they were asked about orientation to the Education Center services, associate degree programs, college credit for military experience, and BSEP. In ranking information requested about career planning, ESOs and counselors indicated the following in descending order of frequency: developing a personal career plan in and beyond the military, assessing interests, and making the transition from military to a civilian job.

ESOs and counselors also responded to a series of questions to determine their attitudes about utilizing computers to deliver ACES information. They indicated that computerization of information about new and existing ACES programs, Department of the Army regulations, master schedule of courses, and MOS and civilian occupations would be useful. ESOs and counselors agreed that a computerized system would provide soldiers with consistent information and would most likely be used frequently by soldiers. They indicated that this type of system would be welcomed by counselors becasue it would enable them to counsel more soldiers by reducing their administrative workload. Counselors also indicated a need for training on the use of a computerized system. In general ESOs and counselors were positive about the value of this type of system as a tool to support Education Center operations.

AREIS Concept. The needs assessment results provided the rationale for AREIS. AREIS is composed of four courseware subsystems. Three of these serve the soldier directly by providing information concerning soldier-selected counseling issues. The fourth subsystem, accessible only by counselors and Education Center staff, stores soldier records and a career database.

Subsystem I, ORIENTATION, is the entry point for the soldier. The objectives of ORIENTATION are to: 1) familiarize the soldier with the computer terminal and printer, 2) provide instruction about the content of AREIS, 3) explain

the Education Center services, and 4) provide an overview of all ACES programs. The Education Center services described in this subsystem include educational counseling and improvement, skill development and recognition, and support services. Brief descriptions of 17 ACES programs including apprenticeship, high school completion, and tuition assistance are available on a menu for selection by the soldier.

The second subsystem, SELF-INFORMATION, is designed to help the soldier define work-related interests, aptitudes, and values. The soldier may select from three on-line assessment instruments which are scored by the computer, based on the soldier's selection of desired education level. The UNIACT® Interest Inventory, designed by the American College Testing Program (1978) provides the soldier with a list of occupations related to his/her interests. The abilities instrument allows the soldier to rate himself/herself on 14 work-related abilities. A list of occupations is generated corresponding to the soldier's self-assessment. The values instrument enables the soldier to examine work-related values and also produces a list of appropriate occupations. The computer generates a summary which consolidates all elements of self-information provided by the soldier.

Subsystem III, GOALS AND PLANNING, helps the soldier identify educational and vocational short-and long-range goals. It also provides details of ACES programs which can help the soldier achieve these goals. The soldier can select from pre-programmed short-range goals which can be attained while serving in the Army such as: Improving Basic Skills, Getting Promoted, and Improving MOS Skills. The soldier may also select from a menu of long-range goals which can be achieved during or after the Army career including: Making a Vocational Choice, Completing the Next Step in Education, and Planning a Military Career.

Subsystem IV, COUNSELOR-ADMINISTRATOR, has been designed to provide counselors with current educational and vocational information to be used during counseling interviews. This subsystem contains local information for each post such as a master schedule of all courses on or near post and points of contact for Army programs. It contains an occupational data file of 420 civilian occupations and an MOS data file which indicates correspondence of MOSs to civilian occupations. Soldiers records are also maintained in Subsystem IV providing a record of each soldier's SELF-INFORMATION assessment and GOALS AND PLANNING activities. It should be noted that this subsystem is accessible only by Education Center staff who have appropriate passwords thus maintaining the security of the soldier's record.

Field Tryout. A field tryout of segments of the four AREIS subsystems was conducted at the Ft. Sill, Oklahoma, Education Center in April, 1980. The following segments were tested:

1. Subsystem I ORIENTATION: an overview of AREIS, Education Center Services, and ACES programs.
2. Subsystem II SELF-INFORMATION: on-line administration of the UNIACT® Interest Inventory.
3. Subsystem III GOALS AND PLANNING: the goal entitled "To Complete the Next Step in Education," designed to provide information about educational offerings on or near post.
4. Subsystem IV COUNSELOR-ADMINISTRATOR: a demonstration of the administrative files which can be maintained by AREIS, including a master schedule of courses and soldier summary report data.

The segments were programmed in PLANIT (Programming Language for Interactive Teaching) on the Army's UNIVAC 1108 computer at the Edgewood Arsenal, Maryland, and delivered to Ft. Sill in a time-sharing mode.

Twelve counselors and 64 soldiers participated in the field tryout. On-line surveys were given to the soldiers prior to using AREIS and after the use of each subsystem to determine their attitudes on the usefulness, clarity, and interest level of all segments. The computer and AREIS content were perceived useful by the soldiers for educational and vocational planning. Counselors indicated that the information provided by AREIS subsystems was useful and accurate. They responded favorably to the delivery of educational information to soldiers by computer.

Also of interest at the field tryout was the operation of the software and hardware. Some limitations of PLANIT were identified, including the inability to: search data files, clear the terminal screen completely, and remain in contact with the mainframe after a five minute delay between users. A cost/benefit analysis was performed to examine alternate delivery systems for AREIS. The hardware configurations compared were a maxi-computer (such as that used during the field tryout), a distributed network of mini-computers, and a stand-

alone microcomputer. The microcomputer was selected for future AREIS development and field test because it has the maximum cost feasibility, requires a minimum of technical and clerical support is easily operated by non-technical personnel, and can be readily installed overseas.

Completion of ARIES Development. The results of the field tryout guided the completion of the four AREIS subsystems. An in-service training package was designed to train counselors and other Education Center personnel in the operation of AREIS. User manuals were written for soldiers and counselors to guide their interactions with the computer as they use AREIS.

AREIS software, programmed in PASCAL, was prepared for use on two multi-user microcomputer systems: the APPLE and the DISCOVERY computers. The APPLE configuration consists of three APPLE II computers with 48K RAM, monitors, a Corvus 10MB Winchester disc, a 5¼" floppy disc drive, a multiplexer and a printer. The DISCOVERY system was selected to test out a second microcomputer system. It consists of a DISCOVERY Basic System (each user has its own 4MHZ Z-80 CPU, 64K RAM and shares a mass storage device of 128 MB), monitors keyboards, and a printer.

EVALUATION

A formative evaluation of the microcomputer version of AREIS is being conducted for nine months in Education Centers at Fort Gordon, Georgia; Fort Meade, Maryland; and Mannheim, Germany. The following questions will be answered during this field test:

1) Does AREIS courseware and software operate as conceptualized?

2) Is the multi-user microcomputer configuration appropriate for Education Center needs?

3) What are the soldier, counselor, and Education Services Officer (ESO) reactions to AREIS and to the use of a computer for dispensing career and education information?

4) What impact does AREIS have on Education Center operations?

The above information is being collected by several means. Each field test site is managed by a local team leader who is responsible for recording problems with AREIS courseware, software, and hardware. Counselors maintain a log of interviews with soldiers which contain the following information: (a) the educational activities of soldiers who used AREIS and those who did not; (b) the interview time with AREIS users and those who did not use AREIS; (c) and the number of soldiers who were invited to use AREIS and declined; and (d) why they declined. Questionnaires will be administered by the computer to assess soldier and counselor reactions to the system.

SUMMARY

ARI is guiding the development and field test of AREIS as one solution to the surge of information which must be provided by a decreasing number of Education Center counselors. The product resulting from the field test will be a tested set of specifications for a microcomputer-based counseling support system designed to meet Army Education Center information needs. The potential payoff of AREIS may be observed in increased soldier potential, job satisfaction and educational growth while supporting the occupational needs of the Army.

REFERENCES

Harris-Bowlsbey, J. & Rabush, C. A needs assessment for the Army Continuting Education System (Research Note 83-17). Alexandria, VA: US Army Research Institute, December 1979.

Katz, M.R. & Shatkin, L. Computer assisted guidance: Concepts and practices (ETS Research Report 80-1). Princeton, NJ: Educational Testing Service, March 1980.

FOOTNOTE

[1]The views, opinions, and/or findings contained in this paper are those of the authors and should not be construed as an official Department of Defense or Department of the Army position, policy, or decision, unless so designated by other official documentation. This project was performed for the US Army Research Institute by DISCOVER/ACT under contracts MDA 903-79-C-0279 and MDA 903-81-C-0569.

COURSEWARE TRANSPORTABILITY ISSUES REVISITED

RICHARD C. BRANDT

UNIVERSITY OF UTAH

ABSTRACT

Courseware transportability issues discussed at the 1982 ADCIS Conference are reviewed in light of the Video-computer Courseware Implementation System group's experience in transporting both lessons and software to a number of mini and microcomputers. The five critical courseware transportability issues discussed are text display, graphics display, data representation, software, and the user environment.

1. Introduction

In 1982, I presented a paper on Courseware Transportability Issues (Brandt, 1982) at the ADCIS Conference in Vancouver. The paper discussed the importance of courseware transportability and recommended the establishment of guidelines for text and graphic display, courseware datafile structures, and software to assure the transportability of courseware. It was argued that if these guidelines were followed, the resulting courseware would be relatively easy to transport. This paper discusses our experiences in following these guidelines in our development of the Video-computer Courseware Implementation System (VCIS) (Knapp, 1983) and in transporting VCIS and courseware implemented using VCIS to other hardware and software systems; specifically, from the Terak 8510a to the HP-9836, the VAX (750 and 780) under Berkeley UNIX, the Gould-SEL, the Terak 8500, the Zenith-100, the IBM PC, and the Victor 9000.

Five critical aspects of courseware transportability will be discussed: text display, graphics display, data representation, software, and environment. The importance of the the environment was unrecognized in 1982. I will discuss the problems encountered, the mistakes made, and the solutions being utilized by the VCIS group at the University of Utah in our current software development project, the Computer-Based Educational Software System (CBESS) for the U.S. Navy.

2. Text Display

In the 1982 paper, the de facto computer industry standard for text display of 24 lines of 80 characters with the top and bottom lines reserved for special prompts and student information was recommended as a standard for computer-based instruction. This de facto standard has worked fairly well; there have been no problems in transporting the text portion of the courseware. Most units have either 24 lines of 80 characters or 25 lines of 80 character screens. The 1982 recommendation that the top and bottom lines of the screen be reserved for system messages and command lines has not been followed. In the CBESS software the bottom two lines are used because it is easier to select a prompt with a digitizer when it is near the bottom of the screen.

VCIS is "frame-based", that is, each frame may be formatted by the author. In contrast, most of the CBESS courseware is "data-based" and the

system determines the format of most of the text so the screen's contents are not predetermined. In this case it is better to view the screen as a large number of definable named windows. Material in this format can be moved to larger screens with little difficulty since only a window redefinition is required,

For text associated with graphics, redefinable windows do not work perfectly. In the CBESS software, the text for graphics labels is usually part of the graphics.

3. Graphics Display

In the 1982 paper, a landscape screen (height = 3/4 width), equal pixel separations in the horizontal and vertical directions, and a minimum resolution of 320 by 240 was recommended for the hardware graphics display. In addition, it was suggested that only RGB screens be used for color, and that video overlay be supported wherever possible. Implicit in the discussion but without specific mention was the support of raster scan rather than vector systems.

Since 1982 we have found that there are a number of subtle differences between the displays. To solve the problems, the following strategy was been adopted for VCIS and CBESS. The graphics software consists of two layers. The user and the stored graphic data only interact with the top layer. The top layer, either the graphics display process or the graphics editor, is machine independent. The top layer obtains from the bottom layer information about the display screen. Using this information, the top layer converts the data into a format appropriate for the screen. The bottom layer, the graphics driver, deals with the screen and the graphics hardware; consequently it is machine dependent and is often written in machine code.

To obtain display independence, we use Bezier control points for all curves and use a virtual screen (+-0.75 in height and +-1.0 width). In 1982 we stored floating point numbers using Digital's format which was not transportable; we now use a BCD format since the IEEE 754 standard for floating point numbers is not yet widely used. When a curve represents a mathematical function, the textual form of the equation representing the function is stored along with the curve's name.

Color presents a number of problems particularly when moving graphics from a system with a large number of colors to a system with fewer colors. Three of these problems and their solutions will be described. First, what happens when an author has selected a color so that it is different from the background and there are now fewer colors? This problem has been solved by introducing a pseudo-color called "different." A different color is displayed as another color when it crosses a background of the same color. When a color is no longer present, it is represented by the "closest" different color. Second, what happens when the author has filled an area to a specified color boundary and the specified color is no longer present? This problem does not have a clean solution; the filling sometimes stops at a boundary before the originally specified boundary. Third, what happens to area fills when a lesson is moved to a system using a vector representation so the screen cannot be probed? In these systems a bit array model of the graphics is maintained in memory. Fortunately memory is cheap and the systems that use vector representations tend to have large amounts of memory. However, the graphic presentation is slowed by the necessary extra computations.

4. Data Representation and Storage

Representing and storing lessons in the form of multiple datafiles rather than programs was advocated in the 1982 paper. Specifically, separate files for graphics, text, animations, and lesson structure were recommended; all the files except the lesson structure file were to be editable. The lesson structure file was to be associated with an editable file, called a command file.

Multiple datafiles have worked well for VCIS. Aside from minor problems with the representation of floating point numbers there have been no problems in moving lessons from one computer system to another. We are restructuring the data structure for the CBESS and new VCIS work to speed up the updating process so that a subject matter expert will be able to see the

effect of changes in a large lesson almost immediately.

The representation of text has evolved slightly. We have added an attribute for determining the color and intensity to the 1982 text descriptors. In 1982 the basic element of text, the text field, consisted of row and column descriptors, a text string, and a terminator. The row and column descriptors are now used only in fixed frame displays. Window and menu displays do not include these descriptors.

The 1982 paper recommended separate files for text, graphics, etc. Although it has worked well, and can be hidden from the user through "Shell" commands (Bourne, 1982); there seem to be advantages in grouping information on a conceptual basis (grouping text, graphics, etc. that is logically related) rather than on a datatype basis. In this format implementation and modification will be faster; however, some redundancy will be present since a copy of a graphics or text frame will occur in a conceptual grouping even if it has been used elsewhere in the courseware.

5. Software

The 1982 paper contained the following five recommendations for software: (1) the editors (text, graphics, animation, and lesson structure) will permit subject matter experts to create and modify the courseware without knowledge of programming; (2) the editors for novices be self-explanatory and follow a "get what you see" format; (3) a hierarchical command structure be used; (4) special unsupported features can be easily added to the system by the user; and (5) the software be written in Pascal.

The first four of the recommendations are still valid today. The only change that is being made for the CBESS software involves the introduction of a single editor for graphics, text, lesson structure, etc. The memory constraints on commonly available microcomputers in 1982 which made this impossible have been largely removed. The fifth recommendation, that the software be written in Pascal, must be changed. Since the Pascal definition (ISO, 1980) does not provide the features needed for system programming, vendors have added their own extensions. The extensions are in many cases incompatible. We handle the extension problem by (1) using macros and a macro preprocessor, (2) eliminating of the use of commonly used Pascal functions such as write and read, and (3) restricting disk access to blockreads and blockwrites, procedures that read and write entire blocks on the disk. Steps 2 and 3 have permitted us to create a layered system; that is, we have isolated the major parts of most programs from both the file system and terminal handlers. For UNIX environments where a sufficiently powerful Pascal was unavailable, a version of the UCSD p-machine has been written in C. This permits UCSD version II programs to run in this environment; however, special features written in machine code and added by users to lessons must be rewritten in C.

Software underlying computer aided instruction must be written in a transportable, powerful language that is suitable for system programming. The only language that meets this requirement now and runs well on microcomputers is C (Kernighan, 1978). Ada meets these requirements and is more readable than C; however, there are very few mainframes and fewer microcomputers running Ada with efficient validated compilers. The CBESS software is being programmed in C; we will be rewriting it in Ada. However, macros will still be used and the major part of most programs will continue to be isolated from both the file system and the terminal handlers.

6. Environment

The 1982 paper did not contain any discussion of the environment since our work had been limited to a single environment.

The current form of VCIS runs in four environments: UCSD Pascal, HP Pascal, Berkeley UNIX, and Microsoft MS-DOS V.2. To reduce our documentation effort and simplify the use, we support a common environment through the use of a shell where possible. The shell provides a common environment for executing programs so that the user sees no difference in executing a VCIS program in any of the environments.

Security issues in a single user environment are very different from those in shared resource, net, and conventional multiuser environments. Techniques that protect student records in the single user environment no longer work. Specifically, in the single user environment, it is reasonable to store a number of student records in a single file. In a multiuser environment such as UNIX, different strategies must be used and additional security measures must be imposed.

7. Conclusion

We conclude that if courseware is written in an environment designed to facilitate transportability, no effort is required to transport the courseware. However, work is required to transport the surrounding software. At present, this work will be minimized by using the language C and layering the software so that only well defined portions depend on the nature of the computer.

REFERENCES

Bourne, S.R. International Computer Science Series. Volume : The UNIX System. Menlo Park, California: Addison-Wesley 1982.

Brandt, R.C. Courseware Transportability Issues, pages 193-196. ADCIS, June, 1982.

International Standards Organization. ISO/DP 7185 - Specification for the Computer Programming Language - Pascal. Pascal News, December 1980, 20, 1-83.

Kernighan, B.W. and Plauger, P.J. The C programming language. Englewood Cliffs, N.J.: Prentice-Hall 1978.

Knapp, B.H and Brandt, R.C. VCIS User's Manual. VCIS Development Group. Salt Lake City, Utah: University of Utah 1983.

COURSEWARE AUTHORING STANDARDS

Carol Clark and Katie Gurney

Data Processing Educational Corp.

ABSTRACT

This talk will discuss the importance of implementing Computer Based Training (CBT) standards for courseware--whether it is produced internally or purchased from an outside vendor. Data Processing Educational Corp. (DPEC) develops CBT courseware for both mainframes and microcomputers. In addition to discussing some of the standards used at DPEC, we will address the following questions:

* Why use standards?
* When should you adopt standards?
* Where should you use standards?
* Who should be responsible for managing the use of standards?
* What are the benefits of using standards?

We will begin with a definition of standards. When we use the term, we are referring to predetermined guidelines that provide the necessary ingredients for quality CBT courseware. Again, even if you're not planning to develop your own courseware, you still need to establish standards to evaluate courseware from CBT vendors.

WHY

Standards are implemented to measure the quality of CBT courseware. Are the courses you are using educationally sound and technically correct? Do they meet not only your CBT needs but also integrate with the overall training strategy of your organization? Implementing standards allows you to establish a model for courseware development.

Your Training Department needs to establish standards for the individuals who will be authoring courseware and insure that the "rules" are followed.

We'll discuss the different phases of a CBT courseware development project and the steps that we follow at DPEC.

WHEN

When should you adopt standards? The answer is at the beginning.

The first step is the Project Plan or Proposal for the CBT course. A good Proposal includes specific sections detailing:

* Deliverables,
* Responsibilities,
* Procedure, and
* Price.

Let's look at each component and discuss some of the standards that apply.

The Deliverable Section presents the overall course design, the final courseware product and any accompanying documentation or student booklets. Provide specific information in this section regarding the final product.

Responsibilities for the development of a CBT course may reside among diverse groups within a company.

First there are Requestors, individuals with a need for a specific CBT course. A Project Leader from this group is designated to obtain approvals throughout the project; to furnish course materials and content for the course; to establish a procedure for review, testing and evaluation of the course as it is being written and also when it is completed.

The Project Leader will be required to interact with an Author--the person responsible for converting the text and content into a quality CBT course. In addition to developing the course, the Author will provide status reports to the Project Leader, meet deadlines and follow the evaluation procedure established by the Project Leader.

A step-by-step procedure provides the necessary structure to complete the project in a timely manner and to everyone's satisfaction. Requestors and Authors will agree on a reasonable time schedule of completion dates.

Although the Project Leader will establish a procedure for review and evaluation of the courseware, the process should be acceptable to the Author as well. For instance, decide that the Author will produce one major revision of a section after student evaluation, not three or four bits and pieces-type changes that can paralyze the completion of the course.

An important item in your Project Plan or Proposal is the price to produce the courseware. If you are considering developing the course internally, you need to carefully analyze the costs: the cost of your Author(s) to create, write and modify the course; the costs associated with the Requestor's group in providing content, testing and evaluating the courseware; the length of time the project will take to complete; the drain on computer resources. After examining these factors, it might be more practical and cost-efficient for you to engage the services of a consultant or an outside company to develop the courseware for you.

In this situation you should arrange a mutually acceptable payment schedule. You don't want to pay "up front" for a course you haven't evaluated and the consultant does not want to wait until completion of the courseware for payment. Break the dollar cost of the course into manageable portions that correspond with due dates for specific sections.

WHERE

Where should you use standards? Let's discuss some of the standards you might want to implement for the Course Design.

At DPEC we divide courseware into an Overview Section, Course Structure Chart and Unit Outlines.

The Overview presents a brief description of the courseware along with the purpose of the training and its place within the overall curriculum. Information pertaining to the appropriate media for the course--remember that CBT enhances and complements other media such as videotape--should be detailed for the student along with the necessary resources to take the course.

The Course Structure Chart depicts, in flowchart-type form, the Units in the course, accompanying manuals and/or prerequisites to the course.

The Course Design also includes specific information about the Unit Outlines such as an overview of material and topics that will be covered. The Units should be further subdivided into objectives or learning units containing a manageable amount of information.

Your Training Department will set standards for the Author to incorporate in the courseware implementation. The standards will include rules on format and content.

A format example:

Courseware naming conventions are useful to identify types of courses. All testing course names can begin with "T" for test; all instruction course names can start with "I" for instruction.

Establish abbreviations where appropriate and maintain them throughout the course and CBT curriculum to insure consistency and eliminate student confusion.

A content example:

In a course Introduction Authors can explain the prerequisites and expected results upon completion. Thus, if the course assumes that the student has an understanding of a CRT terminal, the Introduction can clearly state this so that a student who is unprepared does not become frustrated by the course.

The Introduction can also contain a general description of each Unit, including the topics to be covered and time frame to complete the Unit. Again, always indicate any assumed knowledge so the student has the option to complete any prerequisites.

Authoring standards will resolve many questions like these.

1. What type of layout will be used to develop individual computer screens?
2. Will screens start in column one on the first line and continue all the way to the bottom of the screen?
3. Will a heading appear on each screen?
4. How much information will be presented to the student on each screen?
5. How long will each learning segment be?
6. How often will review occur?
7. When will the student interact?
8. When is reiteration appropriate?
9. Are rhetorical questions useful?
10. When will positive reinforcement be used?
11. Are graphics necessary to teach?
12. When is consistency appropriate?
13. When is variety important?
14. What are the different kinds of questions and when should they be used?
15. Under what conditions is simulation necessary?
16. When is fast-track instruction appropriate?
17. If you are using CBT to test, what are the features of your test bank?

WHO

The question of who manages courseware standards has already been answered: your Training Department or its equivalent.

The point is that courseware standards must be managed by a central administration if you are to reap the benefits.

WHAT

The benefits of courseware standards are a consistent learning experience for your employees that can be measured and evaluated.

The CBT standards that DPEC will present in this talk can serve as guidelines for your organization in addressing and implementing standards. Certainly the ideas discussed will not constitute the only set of standards nor all the CBT standards.

You will experiment and incorporate the ones that meet your training needs and fit in your organization.

Remember, standards provide the necessary structure for creating and evaluating CBT courseware and for measuring its effectiveness as a training medium. Develop your standards, implement and use them consistently and you'll reap the rewards of quality CBT courseware.

INTEGRATING COMPUTER GRAPHICS AND THE VIDEODISC: NEW INSTRUCTIONAL TECHNIQUES POSSIBLE WITH IVIS

Jesse Heines, Lin Olsen, and Roger Bowker

Digital Equipment Corporation

ABSTRACT

Digital Equipment Corporation has developed a new system for integrating computer graphics and the videodisc. This system is called IVIS, the Interactive Video Information System. The unique characteristic of this system is that it allows images generated by the videodisc to be overlaid with computer graphics of industry standard quality (resolution of 960 pixels horizontally by 240 vertically). This quality is achieved by converting the NTSC videodisc signal to RGB rather than using the more common technique of converting the RGB computer signal to NTSC.

The power of this integrated medium offers CAI course developers a number of new techniques for communicating information to students. This paper introduces some of these techniques, emphasizing CAI applications of overlaying. Controlling software is explained, including extensions to a graphics editor for specifying videodisc sequences. An IVIS CAI course on the installation and maintenance of a new printer is described, highlighting applications of the system's unique instructional features.

SOME PROS AND CONS OF VIDEODISCS

As a storage medium for video images, the videodisc is unequalled. Not only can the disc store up to 30 minutes of prerecorded video per side, but each component image of that video can be displayed individually to yield over 54,000 still frames. These capabilities can make the videodisc an important instructional tool for all subjects that exhibit significant visual components.

Even with all of its advanced features and exceptional still frame capability, the videodisc is still not an appropriate medium for all types of visual displays. It is not, for example, a good medium for displaying large amounts of text or fine line drawings, because even the most rock solid image generated by the finest NTSC television cannot match the distinctly crisp lines that can be generated on an RGB computer display of even medium resolution.

Another problem with putting instructional materials on videodiscs is that the discs cannot be altered once they are pressed. (There has been much talk about "write-once" discs, but they are not yet economical.) This characteristic can make it difficult to revise instructional materials once they are tested, and to correct errors that pop up only after the disc has been in use for some time.

SOME PROS AND CONS OF COMPUTER GRAPHICS

Computer graphics, on the other hand, offer excellent resolution for many types of line figures. However, the natural images that are easily displayed from a videodisc are very difficult to produce by computer. While some computer graphic systems allow limited animation, the types of systems commonly used for CAI only allow the instructional designer to

animate very small entities, such as a few special characters. Full screen motion of the type we are used to seeing on television is virtually impossible on systems commonly used for CAI today.

Unlike the videodisc, computer graphics are particularly well suited to displaying text on video screens in a large number of sizes, fonts, directions, and colors, with special attributes such as blinking, bolding, and reverse video. Most experienced CAI screen designers would agree that one should almost always use as little text on the screen as possible (illustrative diagrams are the preferred communication vehicle). When text is used, however, the screen resolution must be fine enough to allow that text to be read and understood easily.

Graphics stored on computer disks (or systems) are generally easy to update. If the graphic code itself cannot be patched, it is usually easy and inexpensive to generate new disks and distribute these to the system's users. This feature allows instructional designers to test their materials in final form and make revisions based on actual student usage. Most storage media for graphics software also make it possible to correct mistakes that are discovered only after the product has gone to the field.

THE BEST OF BOTH WORLDS

Digital Equipment Corporation has developed an integrated videodisc and computer graphics system that offers the best qualities of both of these media. This system is called IVIS, the Interactive Video Information System. The heart of the system is an option to the Professional 350 minicomputer that contains:

- a circuit for converting the NTSC signal generated by the videodisc to an RGB signal compatible with that produced by the Professional 350.
- a circuit that allows the video signal generated by the computer and that produced by the videodisc to be mutually synchronized, and
- a circuit for mixing the two synchronized RGB signals into a single signal (960 pixels horizontally by 240 pixels vertically) and displaying it on a medium resolution RGB color monitor.

The complete IVIS system also includes the base Professional minicomputer (including floppy diskettes, 10Mb of hard disk, floating point processor, and color graphics system), a computer floor stand, a videodisc player using an RS232 interface, a Digital VR241 color monitor (capable of displaying up to 8 colors simultaneously from a palette of 256), and optional printers.

The Professional 350 IVIS option can operate in four modes, each with a unique application in course presentation.

- Professional 350 mode -- all capabilities of the Professional 350 are available to the developer (including lower resolution/greater color selection). However, when the IVIS option is switched from this mode, the screen will roll until synchronization is acquired.
- Synchronized Professional 350 mode -- all eight colors are available to the user and display from the videodisc source is not possible. This mode is used for extended computer-based instruction sequences that wish to use the eighth color but that need to convert to IVIS mode without the roll associated with the first acquisition of synchronization.
- Video mode -- the signal coming from the video source is displayed directly on the monitor with no switching between graphic and video sources. This mode is used when all the information is coming from the videodisc.
- IVIS mode -- one of the eight colors (selected from the 256-color palette) is designated as transparent, allowing both graphic and video images to be displayed on the screen simultaneously.

In IVIS mode, the screen can be completely devoted to the videodisc by setting the background color to transparent. With the screen background set to any other color, only the computer-generated text and graphics are displayed. To overlay computer graphics onto a videodisc image, the screen background is first set to transparent, allowing the video image to show through. Other graphics are then drawn on the transparent background in any of the remaining seven colors, effectively blocking out the video at those points and simultaneously displaying their information.

THE IVIS HARDWARE

A simplified block diagram of IVIS is shown in Figure 1. Experiments with converting the computer-generated signal to NTSC resulted in intolerable visual quality. Colors bled into each other and shifted in hue and saturation, and the system was not able to display lines with single pixel widths. Industry standard text displayed 80 characters per line was illegible. Digital's engineers therefore converted the NTSC videodisc signal to RGB even though this approach is more difficult and expensive (though straightforward).

The real difficulty in building a system of this type is in the synchronization of the two video signals. Computer terminals and computer graphics subsystems are virtually never designed to be synchronized to external sources. With industrial quality videodiscs or video tape, it is reasonable to vertically synchronize the video source to the computer. However, if horizontal synchronization with an accuracy of less than one-third of a pixel is desired, a timebase corrector would be needed. It becomes a more attractive alternative to synchronize the computer to the video source than to require the expensive equipment needed to do it the other way. IVIS chose this route.

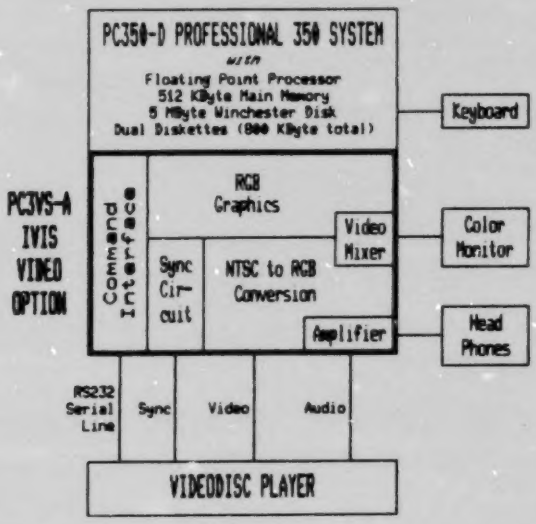


Figure 1

IVIS BLOCK DIAGRAM

Even though the visual qualities of IVIS graphics and video are both excellent individually, the system's "pièce de résistance" is its ability to overlay the former on the latter. IVIS implements overlay like a chroma key. That is, one of the eight available Professional 350 colors is defined as transparent. (In non-mapped applications the transparent color is black, in mapped applications, color 0 is transparent and the other 7 colors can have any value selected from a palette of 256.) The video switcher compares each pixel from the bit-mapped memory to the transparent value as it is taken from the memory for display. It displays the contents of the bit-mapped memory for each non-zero pixel. When it finds a transparent pixel, it displays the corresponding value from the videodisc for a pixel's duration. With a bit map of 960 by 240 pixels being refreshed at 60 times per second, this means that the decision about which video source to display is made approximately every 60 nanoseconds.

The ability to overlay was an immutable design goal from the beginning of the project. Instructional designers conceived of a large number of applications for overlaying, and felt that this feature was critical to the system's success. Some typical uses of overlaying are:

- highlighting specific components in shots of a complex device by outlining or pointing to them,
- drawing hidden data paths to show how pictured components interconnect,
- adding orientation information at the top and bottom of the screen,
- "lighting" different LEDs on the same shot of a control panel while the meanings of these LEDs are described,
- providing student control features on the screen during a video sequence,
- using composite video images as menus and allowing students to indicate their choices by moving the cursor to different areas of the image,

- using the video image to hold a number of smaller images and displaying only those applicable to the subject being discussed,
- implementing exercises that ask students to identify or sequence specific components by pointing to them, and
- correcting slightly erroneous video information by blanking out the incorrect screen area, overwriting the error, displaying explanatory text, or programming around the error and ignoring the video.

THE IVIS SOFTWARE

The entire system is, of course, tied together with controlling software. The approach to the software problem was to extend the graphics editor described at the 1980 ADCIS Conference [1]. This software allows a developer to control screen color easily and to generate graphics using editor commands or GIGI's firmware graphics language, ReGIS. The graphics editor has also been extended to allow control of the videodisc. About a dozen such commands have been implemented, the main ones being:

- **LOAD** to spin up the videodisc and establish communication between the player and the GIGI,
- **FIND** n to search to videodisc frame number n while blanking the screen,
- **PLAY TO** n to play the video segment between the current frame and frame n and then pause on frame n (playing can be either forward or backward),
- **AUDIO [1|2] [ON|OFF]** to turn the two audio channels on and off, and
- **WAIT** to cause the host system to cease further output to the terminal until it receives a signal from the videodisc indicating that playing of the last requested videodisc segment has been completed.

The IVIS hardware works with any NTSC video source, but non-videodisc sources would require different controlling software. Examples of the ways in which these commands can be used and integrated with computer-generated graphics are presented in the next section.

INSTRUCTIONAL TECHNIQUES

The first course developed for IVIS was designed to introduce DEC Field Service Engineers to a new series of printers. These printers are similar in some respects to printers with which the engineers should already be familiar, but they have a number of new components and features. Due to the prior experience of the target population and the existence of well-documented troubleshooting job aids, it was felt that students could learn all they needed to know to install and maintain these printers without actually working on them, providing they had adequate instruction. These factors and the strong visual component of the device itself led to the selection of this course as the first real IVIS application.

The course begins with an overview of the printer series, pointing out the salient features and describing the major components. This overview provided the opportunity to use straightforward techniques to highlight specific components as they were being described on the audio track. A picture of the entire device was displayed, and each component was either outlined or pointed to as it was discussed. At times part of the videodisc picture was masked to direct the student's eye to the relevant components. This was done by overwriting part of the transparent background with a block of a different color, usually black. Note that black, the normal dark background color of the screen, is a valid color just like any other. It still blocks the videodisc signal effectively even though it appears that nothing has been written.

Once an entire set of components had been described, it was often valuable to display a diagram showing how they interacted. An audio track to serve this function was recorded on the videodisc, but corresponding video had not been shot. The plan was to draw the diagram solely with computer graphics and play the videodisc audio track while the diagram built up on the screen. Using the WAIT function described above, display of the diagram could be synchronized to the audio track. (The graphics generally came up faster than the narration that was describing them.) This use of sound alone proved to be so effective that computer graphics were often substituted for video sequences that actually were on the disc but that were to be of inferior quality. Note that by using the AUDIO command described above to switch from

one audio track to the other, course developers were able to make use of the entire 60 minutes of audio available on one side of the disc.

One requirement was to show actual video of a pair of switches so that students could locate them on the printers, and then simulate various switch positions as an exercise. In this instance, video showing the switches was first overlayed with computer-generated graphics, and then the switches were redrawn in a larger size. This technique allowed visual continuity to be maintained after orienting the student to the component at hand. Again, IVIS made it possible to take advantage of the best features of both the videodisc and the computer graphics media.

Several video menus were planned for the course. These consisted of nine blocks arranged on the screen in three rows of three. The eight peripheral blocks were filled with pictures of various printer components, and the student was directed to use the terminal's arrow keys to move the cursor to the picture of the component he or she wanted to study. (The middle block contained other options, such as returning to the main course menu and exiting the course.) The matrix of video pictures was composed with standard video editing equipment and stored on a single frame of the videodisc. Using the FIND command described above, it was easy to get to this frame whenever the menu was desired.

One of the most fundamental tasks that Field Service Engineers had to learn about the new printers was how to assemble and disassemble them. This was presented with a standard video sequence, but we thought that retention would be enhanced if students could step through the process under their own control. We therefore implemented an exercise using the STEP command described above that worked as follows. A picture of the fully assembled printer was displayed. At the left of the screen a list of parts was displayed. Each part name was printed in green. To remove the next part in the disassembly procedure, the student pressed the down arrow key. The videodisc advanced to the frame showing the printer with this part removed, and the color of the part name was changed from green to red. The student could continue removing parts in this manner or put parts back by pressing the up arrow key. He or she could spend as much or as little time as needed to gain familiarity with each of the parts and their assembly and disassembly procedure.

ADDITIONAL TECHNIQUES TO EXPLORE

The techniques described above represent a fair sampling of the IVIS capabilities explored to date. Some techniques work better than others, but it appears that all techniques have applicability in certain unique situations.

The videodisc for this course was pressed before the computer materials were developed. While video was shot to allow for a large number of combination techniques (such as the menus described above), new techniques were created when some computer materials proved difficult to implement because the video had not been shot with those techniques in mind. For example, it might have been nice to have more close-ups of the switches described in the previous section so that the simulation could have been done by overlaying on actual pictures rather than redrawing the switches. Closer attention could also have been paid to the angles from which the printer was shot to achieve greater visual continuity between those images and the computer-generated diagrams.

All-in-all, this attempt at using IVIS turned out to be an extensive learning experience. More up front planning is being done on courses currently in development, as the parameters that are most likely to get out of control during the development cycle have now been identified.

NOTES

[1] Moreau, Ken, and Jesse M. Heines, 1980. "Creating Graphic Displays on Non-Graphics Terminals," 1980 ADCIS Conference Proceedings.

COMPUTER BASED EDUCATION:

PREPARATION OF AN EDUCATIONAL TECHNOLOGIST

D. L. Jelden

University of Northern Colorado

ABSTRACT

With the application of computers in education to control the management and testing, video presentations and tutorial lessons administered to students, the preparation of individuals who are to work with this type of technology needs clarification. This presentation will discuss the varied educational background, required technical skills, computer competencies and theoretical base, essential for the preparation of 'educational technologists'. The more sophisticated the technology available to the relationships of technology to the classroom teacher, the greater the demand for an understanding of the pedagogical strengths, weaknesses, advantages and disadvantages for instructional purposes. Knowing the best technology to employ for the job at hand will be a commonplace demand for the classroom teachers. Discussion will also center on the major academic disciplines involved in preparing educational technologists and the organization of specific course offerings for such a program.

INTRODUCTION

The traditional educational program with its parochial approach in specific disciplines is become less relevant in a society which is becoming more and more technological. As society becomes more complex and more dependent upon technology in the educational marketplace, greater emphasis needs to be made in the interdisciplinary nature of learning blending both technology and education. The transition toward a more 'Renaissance' type of person capable of teaching and working in a technical environment within the public and private school system is becoming more apparent. The availability of 'educational technology' and the application of the computer as both manager and tutor requires that all teachers be schooled in the general nature of this instructional technology as a support system to their academic discipline. Further, the development of specialized educational technologists is also at hand. Knowing the strengths, weaknesses and characteristics of technology available to education will improve the teaching/learning process in any discipline.

WHAT IS EDUCATIONAL TECHNOLOGY

There seems to be no specific agreed upon definition of 'educational technology'. Each individual defines it in relation to their own background and experience. The media specialist uses audio/visual hardware as a point of departure, the educational psychologist uses testing/student analysis as an emphasis and the computer scientist uses hardware/software capabilities as their starting place. To a generalist in the field it is obvious that all of these areas are a part of the educational technologist's background.

Educational technology is a diverse, multi-disciplinary program of studies. The utilization of knowledge in many disciplines, such as electronics, group and individual communication, psychology, administration, research, educational media, and computer science is essential to an educational technologist's training. An educational technologist is one who pulls the theoretical base of several disciplines into an applied setting. As the name suggests, the technologist is a

practitioner of education who utilizes the theory and technology of many disciplines and pools that knowledge to solve teaching/learning problems. A working definition of an educational technologist could be described an an individual who uses 'the best we know in theory and practice synthesized toward solving educational problems.'

ADMINISTRATIVE STRUCTURE

In order to provide for the diverse experiences essential to the preparation of educational technologists, the removal of restricting bonds inherent to a unique or single discipline of studies is needed. The traditional program in professional education is not capable of providing all of the theoretical and technical expertise required for a competent educational technologist.

Some undergraduate and most graduate program offerings in existing colleges and universities are not geared for providing much interdisciplinary activity. Each has a direct accountability relationship to its own discipline. Provision needs to be made to diversify undergraduate and graduate offerings in order to better serve the diverse training needs of the educational technologist.

Because any interdisciplinary program has no "home" and thus no direct accountability relationships, it is imperative that some type of program responsibility is identified. For undergraduate programs, administration of the program may be assigned to the school of education. At the graduate level, the graduate school may become the administrative agent. The autonomy reflected by most existing graduate programs within a given discipline impede the entrance of 'outside' students into those disciplines. The attitude of most academic disciplines is to "take care of majors" first and then provide for outside students as time and class space become available. This attitude is understandable. Beyond this issue, however, is the imposition of requiring massive undergraduate prerequisites for non-major entrance into the graduate programs of some disciplines. This sometimes unnecessary requirement discourages enrollment in advanced classes by 'outside' students. To keep these type restrictions at a controllable level and to establish a working relationship for graduate programs at the interdisciplinary level, the graduate school appears to be in a crucial and most advantageous position. They seem to be in the best position to coordinate interdisciplinary programs at the graduate level.

EDUCATIONAL TECHNOLOGY PROGRAMS

Existing course offerings and graduate programs of most accredited institutions have access to the theoretical and technical course content to prepare educational technologists. Because of the professional and technical experiences required, these courses are not, however, generally housed in a single academic discipline. It, therefore, is apparent that a program of studies which is interdisciplinary in nature is the most practical way to prepare educational technologists. A single administrative unit, be it psychology, education, computer science, technology or communications is not able to provide all of the competencies required of the educational technology graduate. The program should be institution wide, not department oriented. Program advisement, however, should probably be done within the professional educational program.

To provide for specific examples of the interdisciplinary nature of such a program, here are a collection of courses recommended in one institution which are used to produce an educational technologist. No attempt is made here to say this list is totally complete or that the course offerings are totally adequate. This example is given to demonstrate the variety of subject fields involved and the types of skills needed by the educational technologist. It also demonstrates the interdisciplinary nature of the course offerings. These course titles are listed under three major blocks--Instruction, Research and Service/Evaluation, the major areas of technical competence required in the program.

	INSTRUCTION
Discipline	
Technology	Technology and Its Impact on Society
	Technical Illustration/ Computer Graphics
	Maintenance of Educational Technology Hardware
	Digital/Microprocessor Basics
	Analog Electronic Devices
	Construction Techniques (Simulators/Dynamic Demonstrators)
Education	Orientation to Computer Based Education

	Computer Managed Instruction
	Computer Assisted Instruction
	Educational Analysis Techniques
	Individualized Learning Activity Packages
	Design/Construction of Audio Visual Resources
	Cataloging/Classification of Educational Materials
	Philosophy/Theory of Instructional Technology
	Educational Accountability
Administration	Introduction to Administrative Theory
	Educational Planning
	Management Systems Analysis
	Educational Decision Making
	Executive Leadership
Communications	Photography (Black and White/Color)
	Basic Radio Production
	Television Production
	Cable Television/ Satellite Transmission
	Organization/Production in Graphic Arts
	Offset Lithography
	Teleconference Principles
Psychology	Principles of Learning
	Human Information Processing
	Theories of Motivation
	Psychology of Individual Differences
	Learning Theories: Cognitive Approaches
	Psychological Services
	RESEARCH
Research	Introduction to Research
	Educational Storage and Retrieval Systems
	Research Proposal Preparation
	Evaluation of Research
	Research Methodology
	Educational Test and Measurements
Statistics	Descriptive Statistics
	Inferential Statistics
	Analysis of Variance
	SERVICES/EVALUATION
Evaluation	Evaluation and Measurement
	Evaluation of CBE Software
	Hardware Evaluation and Interfacing
	Educational Systems Evaluation
Computer Science	Programming Languages
	BASIC
	FORTRAN
	COBOL
	ADA
	APL
	PILOT (Super)
	LOGO
	CAI (Tutorial Systems)
	PHOENIX (Mainframe Menu Systems)
	Computer Languages (Machine, Assembler, Compiler, Interpreter)
	Networking Computer Systems
	Video-Disk Interfacing
	Data Bases (Interfacing and Access)
	Data Processing Systems/ Methods

From the collection of courses just presented, you can see that nine disciplines are required to meet the educational preparation of the requirements. Although some of the courses listed might be available in the undergraduate program, it is essentially a graduate level offering. This implies that people entering the educational technology field probably have a subject area specialization in their undergraduate program and a degree in education with a teaching credential. These are considered basic for entry level into an educational technology degree program. People without an undergraduate education degree would have deficiencies in educational psychology and educational foundations which would need strengthening.

FUTURE EXPECTATIONS

Traditional educational media programs in higher education have addressed the development of instructional audio/visual aids for the classroom teacher. With the advent of the computer and its baby brother, the micro, a whole new generation of instructional devices

and educational systems became available. The ability to interface a number of different types of media and a higher level of sophistication in the teaching/learning process have permanently altered the traditional way teachers teach and the avenues available for students to learn. Educational Technology is not a glorified souped-up version of educational media but is an extensive new direction open to education. The interfacing of student assessment, study habit skills, establishing learning/teaching styles with new and expanding technologies is the future of education. The teachers' activities and classroom responsibilities will, of necessity, have to be changed. The decentralization of information, the extensive utilization of technology planned for the future will change the role of the teacher from presenter to facilitator. The teacher of tomorrow will be a user of technology in the classroom and must be informed of the affects that technology will have on them and their students. Having an educational technologist available for consultation will provide support to that classroom teacher and relieve them of the excessive pressure generated by new high levels of technology.

ROBOTICS: TRAINING CHALLENGE OF THE 1980s?

B.K. Lindauer, Denver Research Institute

L.J. Bruemmer, FARED Robot Systems

ABSTRACT

THE TREND TOWARD INCREASED USE OF ROBOTICS IN THE INDUSTRIAL WORK PLACE IS PREDICTED TO HAVE A DIRECT IMPACT ON TRAINING DEVELOPMENT, INCLUDING COMPUTERIZED FORMS OF INSTRUCTION, WITHIN THIS DECADE. BY THE YEAR 1985, 50,000 ROBOTS MAY BE IN USE BY AMERICAN INDUSTRY WITH AN ANNUAL PROJECTED GROWTH RATE VARYING BETWEEN 30 AND 49 PERCENT. CONSERVATIVE ESTIMATES INDICATE THAT A MINIMUM OF 20,000 WORKERS WILL BE AFFECTED IN THE IMMEDIATE FUTURE. ALTHOUGH THESE WORKERS WILL COME PRIMARILY FROM ASSEMBLER, CHECKER, PAINTER, WELDER, AND MACHINIST CATEGORIES, TWO DIFFERING VIEWS HAVE BEEN EXPRESSED REGARDING THEIR DISPLACEMENT. ONE VIEWPOINT SUGGESTS THAT THE MOST SIGNIFICANT CHANGES TO THE WORKER WILL BE FEWER DEMANDING, DANGEROUS, OR REPETITIVE JOBS COUPLED WITH INCREASED TRAINING REQUIREMENTS FOR NEW JOBS. THE ALTERNATIVE VIEW PREDICTS WORKER DISPLACEMENT BY JOB ELIMINATION WITH LIMITED OR NO OPPORTUNITY FOR RETRAINING IN THE ORIGINAL WORK PLACE. IN EITHER CASE, NEW TRAINING OR RETRAINING BECOMES AN ISSUE FOR INDUSTRIAL TRAINERS AND ADMINISTRATORS. THIS PAPER EXPLORES THE RELATIONSHIP BETWEEN PREDICTED DISPLACEMENT OF WORKERS DUE TO IMPLEMENTATION OF ROBOTICS TECHNOLOGY, THE ABILITY OF INDUSTRY TO DEVELOP EFFICIENT AND EFFECTIVE TRAINING TO MEET CHANGING JOB REQUIREMENTS AND POTENTIAL CONTRIBUTIONS TO TRAINING TECHNOLOGY FROM THE ROBOTICS FIELD.

OVERVIEW

THE IMPACT OF ROBOTS ON AMERICAN INDUSTRY IS A TOPIC THAT HAS BEEN RECEIVING INCREASED ATTENTION RECENTLY, BOTH FROM THE MEDIA AND FROM SCIENTIFIC/TECHNOLOGICAL SOURCES. THE FOCUS ON ROBOTICS HAS, IN PART, BEEN FOSTERED BY THE MEDIA'S PROJECTIONS OF THE INTELLIGENT CAPABILITIES OF ROBOTS (I.E., REPLACEMENTS FOR MANY HUMAN FUNCTIONS) AS WELL AS GENERAL CONCERNS FOR THE IMPACT OF RAPIDLY DEVELOPING TECHNOLOGIES ON AN ALREADY HIGH UNEMPLOYMENT RATE. THE PUBLIC PERCEPTION OF THE INDUSTRIAL ROBOT AS BEING ALMOST "HUMAN" MAY HAVE CONTRIBUTED TO THE VIEW THAT ROBOTS ARE BOTH QUANTITATIVELY AND QUALITATIVELY DIFFERENT THAN OTHER FORMS OF AUTOMATION TECHNOLOGY AND WILL HAVE A MAJOR AND IMMEDIATE IMPACT ON INDUSTRY. THE PURPOSE OF THIS PAPER IS TWOFOLD: FIRST, TO EXPLORE THE POTENTIAL EFFECTS OF THE USE OF INDUSTRIAL ROBOTS ON THE WORK PLACE, AND SECOND, TO EXAMINE THE IMPLICATIONS FOR TRAINING, BOTH FOR TRAINING OF NEW WORKERS ENTERING THE LABOR FORCE FOR THE FIRST TIME AND RETRAINING OF WORKERS WHO HAVE LOST JOBS AS THE RESULT OF AUTOMATION.

THE USE OF ROBOTS BY INDUSTRY HAS BEEN JUSTIFIED ON THE BASIS OF THEIR ABILITY TO INCREASE INDUSTRIAL PRODUCTIVITY AND, AT THE SAME TIME, IMPROVE U.S. COMPETITIVENESS IN THE INTERNATIONAL MARKETPLACE. ROBOTS, ALONG WITH RELATED FACTORY AUTOMATION TECHNOLOGY, ARE VIEWED AS HAVING THE CAPABILITY TO LOWER PRODUCTION COST, IMPROVE OVERALL QUALITY AND RELIABILITY OF MANUFACTURED PRODUCTS, AND REDUCE OR ELIMINATE REPETITIVE AND HAZARDOUS JOBS

FROM THE WORK ENVIRONMENT (OFFICE OF TECHNOLOGY ASSESSMENT, 1982). THE MOST COMMONLY CITED EXAMPLE OF THE SUCCESSFUL APPLICATION OF INDUSTRIAL ROBOTS IS THE JAPANESE AUTOMOBILE INDUSTRY WHERE THEY ARE USED PRIMARILY IN WELDING AND PAINTING APPLICATIONS. ROBOTS IN JAPAN ARE ATTRIBUTED WITH NOT ONLY IMPROVEMENTS IN PRODUCT QUALITY, BUT ALSO WITH REDUCED LABOR COSTS.

WITH INCREASING EMPHASIS ON PRODUCTIVITY, AMERICAN INDUSTRY HAS BEEN FORCED, BY THE DEVELOPMENT OF ROBOTICS AND AUTOMATION TECHNOLOGY, TO EXAMINE AND STREAMLINE ITS ENTIRE MANUFACTURING PROCESS. AS A RESULT, ROBOTS ARE SEEN AS AN INTEGRAL PART OF THE REVITALIZATION OF THE INDUSTRIAL WORK PLACE. ESTIMATES VARY, BUT APPROXIMATELY 50,000 INDUSTRIAL ROBOTS ARE PROJECTED TO BE IN USE IN THE UNITED STATES BY 1985, WITH AN ANNUAL PROJECTED GROWTH RATE VARYING BETWEEN 30 AND 49 PERCENT (HUNT, 1983). THE RATE OF GROWTH WILL BE DETERMINED IN PART BY THE ROLE OF TECHNOLOGICAL CHANGE IN SUCH AREAS AS COMPUTER-ASSISTED DESIGN; COMPUTERIZED INFORMATION SYSTEMS, INCLUDING ARTIFICIAL INTELLIGENCE; AUTOMATION TECHNOLOGY, AS WELL AS ROBOTICS; AND THE PATTERN OF TECHNOLOGY DIFFUSION THROUGHOUT INDUSTRY (CONDON, 1983). DESPITE THE PERCEIVED ADVANTAGES OF THE USE OF ROBOTS, MAJOR IMPEDIMENTS TO LARGE-SCALE IMPLEMENTATION AS VIEWED BY THOSE CURRENTLY ACTIVE IN THE FIELD ARE ECONOMIC JUSTIFICATION, RESEARCH AND DEVELOPMENT REQUIREMENTS IN THE SOFTWARE/PROGRAMMING AND HARDWARE AREAS, AND TECHNICAL PERSONNEL REQUIREMENTS (SOCIETY OF MANUFACTURING ENGINEERS, 1982). THE NATIONAL BUREAU OF STANDARDS PROJECTS THAT UNLESS TECHNOLOGICAL CHANGE AND DIFFUSION OCCUR AT A MUCH FASTER THAN ANTICIPATED RATE, ROBOTS WILL NOT HAVE A SIGNIFICANT IMPACT ON OVERALL NATIONAL PRODUCTIVITY UNTIL 1990 (ALBUS, 1982). HOWEVER, PRACTITIONERS IN THE FIELD VIEW THE 1980S AS A TIME OF RAPID TECHNOLOGICAL CHANGE WITH RAPID MOVEMENT TOWARD FACTORY AUTOMATION AND PLANT INFORMATION SYSTEMS.

THE EFFECTS OF ROBOTICS ON THE LABOR FORCE HAVE BEEN LARGELY RESTRICTED TO THE UNEMPLOYMENT ISSUES. DUE TO RAPID TECHNOLOGICAL CHANGE, HIGHER THAN NORMAL WORKER ATTRITION RATES ARE PREDICTED FOR MANUFACTURING INDUSTRIES. THIS HIGHER UNEMPLOYMENT RATE CAN THEN BE COMPARED TO AN EXPANDING NEED FOR SKILLED TECHNICIANS TO SERVICE THE NEW INDUSTRY. THE LACK OF DEFINITIVE DATA ON WORKER DISPLACEMENT (ELIMINATION OF PARTICULAR JOBS) DUE TO ROBOT IMPLEMENTATION HAS CREATED TWO CONFLICTING VIEWS: ONE VIEWPOINT SUGGESTS WORKERS WILL EXPERIENCE FEWER UNDESIRABLE JOBS, COUPLED WITH INCREASED TRAINING REQUIREMENTS FOR NEW POSITIONS; THE OPPOSING VIEW FOCUSES ON WORKER DISPLACEMENT WITH LIMITED CHANCE FOR OBTAINING NEW EMPLOYMENT. BEFORE THE ISSUE OF ROBOTICS' IMPACT ON INDUSTRIAL WORKERS CAN BE RESOLVED, INDUSTRY ANALYSTS, INCLUDING ADMINISTRATORS AND TRAINERS, MUST FIRST UNDERSTAND THE CAPABILITIES OF THE CONTEMPORARY ROBOT.

THE ROBOT INSTITUTE OF AMERICA (RIA) DEFINES A ROBOT AS FOLLOWS:

> A ROBOT IS A REPROGRAMMABLE MULTIFUNCTIONAL MANIPULATOR DESIGNED TO MOVE MATERIAL, PARTS, TOOLS, OR SPECIALIZED DEVICES THROUGH VARIABLE PROGRAMMED MOTIONS FOR THE PERFORMANCE OF A VARIETY OF TASKS.

THE RIA FURTHER SUBDIVIDES ROBOTS INTO FOUR CATEGORIES: VARIABLE SEQUENCE (OPERATIONAL STEPS CAN BE CHANGED), PLAYBACK (OPERATIONS ARE EXECUTED FROM MEMORY), NUMERICAL CONTROL (CONTROL SOFTWARE BASED ON NUMERICAL DATA), AND INTELLIGENT (SENSORY PERCEPTION AND DECISION MAKING CAPACITITES). JAPANESE MANUFACTURERS INCLUDE TWO ADDITIONAL CLASSES OF MANUAL MANIPULATOR (OBJECTS MANIPULATED BY OPERATOR) AND FIXED SEQUENCE (REPETITIVE OPERATIONS THAT ARE NOT EASILY CHANGEABLE) ROBOTS. MOST OF THE ROBOTS CURRENTLY IN OPERATION ARE PRIMARILY SIMPLE MANIPULATORS EMPLOYED IN WELDER, ASSEMBLER, CHECKER, PAINTER, AND MACHINIST CATEGORIES, WITH RELATIVELY FEWER NUMBERS IN THE SOPHISTICATED OR INTELLIGENT CLASSES THAT ARE CAPABLE OF SENSORY PERCEPTION AND DECISION MAKING. THE DIFFERENCE IN DEFINITIONS BETWEEN COUNTRIES IS IMPORTANT WHEN DEGREE OF TECHNOLOGY IMPLEMENTATION IS CONSIDERED. BY 1980, THE JAPANESE HAD AN ESTIMATED 76,500 ROBOTS IN OPERATION ACCORDING TO THEIR MORE INCLUSIVE CLASSIFICATION. WHEN THE MORE RESTRICTIVE U.S. DEFINITION WAS APPLIED, THE NUMBER WAS REDUCED TO APPROXIMATELY 11,000.

IMPLICATIONS FOR TRAINING

Given the state of robotics technology and its current and projected applications in industrial manufacturing, continuous change and adaptation for several levels of industrial training are implicated in order to accommodate the labor force. Current research indicates that labor within the automobile industry will be influenced first by robotics/high technology automation since 25 percent of U.S. robots are installed there. Conservative estimates predict that, for the State of Michigan alone, 13,500 to 24,000 jobs will be lost to robots. At the same time, however, the robotics industry is predicted to create somewhere between 5,000 to 18,000 new jobs within the state. The newly created jobs, however, require more high-level skills than the jobs they replace, resulting in a skills gap (Hunt & Hunt, 1983). In a recent (1982) Delphi forecast by the Society of Manufacturing Engineers (SME), panelists projected that approximately 20,000 workers will be affected by industrial robots within this decade, compared to 70,000 to 100,000 new positions created by the robot industry. Only 6 percent of these affected workers are expected to be terminated. This figure will subsequently drop to 2 percent. Twenty-five percent of the displaced workers are predicted to be transferred without training, 50 percent will receive new training for positions in the same plant, 13 percent will be retrained for transfer to a new plant, 6 percent will be retired early, and 6 percent will be terminated. This figure contrasts sharply with Bluestone and Harrison's (1982) prediction that robots will knock out five jobs for every one that they create.

Anecdotal evidence from the Japanese industries that have automated using robots initially reported less resistance to robots because of labor practices, specifically lifetime employment, company-wide unions, and extensive training programs (Aron, 1982). However, recent opposition from unions has been noted. For example, at one of Nissan Motor Company's factories, 70 percent of all workers have been reassigned at least once, supervisors in some instances have been replaced by skilled technicians, and low-level supervisors have been demoted. Union leaders now are on record opposing the introduction of robots (Lee & Zemke, 1983). The UAW has also recognized that the long-term effect of technological change, including use of robots in factory automation, is a reduction in nonskilled labor and has issued guidelines for those involved in the design and construction of robots/automation (Weekly, 1979-80). These guidelines suggest that on-the-job training be provided so that existing skilled workers can be utilized.

Regardless of which predictions are borne out, several factors remain that will influence future training programs: (1) worker displacement, without attaching degree or numbers, will occur as factories automate and implement the use of robots; (2) retraining of some workers will become a necessity; (3) there will be a gap in skills between existing workers and required new technicians; and (4) workers as well as managers and supervisors will be influenced by changing social systems in the work place as a result of automation.

Management Training

Before actual work on the implementation of robots is begun, management at a variety of levels must be aware not only of robot capabilities in improving productivity, but also their potential impact on human relations throughout the factory. In a recently published anecdote, one engineer is reported to have said that failure of robotic applications was due to 5 percent robot-related problems, 0 percent worker problems, and 95 percent due to management/supervisory indifference (Clapp, 1981). A contributing factor to lack of management support may be unrealistic expectations of the capabilities of contemporary robots. In other words, there may be an information gap between user knowledge of what is currently available and what is predicted for the future. A past president of RIA emphasized the need for the robotics industry to communicate with top management, since these individuals would make the ultimate decision regarding the implementation of new technology and, therefore, should be the targets for increased educational efforts (Stouffer, 1980-81). At a

MINIMUM, UPPER MANAGEMENT MUST BE FAMILIARIZED WITH STATE-OF-THE-ART TECHNOLOGY, RETURN ON INVESTMENT/ECONOMIC JUSTIFICATION, QUALITY IMPROVEMENTS (PRODUCTIVITY); CURRENT, RELIABLE ROBOTIC APPLICATION AND LABOR IMPACTS. THE RIA HAS BEEN INSTRUMENTAL IN PROVIDING TRAINING AT THIS LEVEL THROUGH MEMBERSHIP IN ITS CORPORATE USER GROUP AND THROUGH SPONSORING TIMELY WORKSHOPS FOR INTERESTED SEGMENTS OF THE INDUSTRIAL COMMUNITY.

HOWEVER, TRAINING AND EDUCATION OF LOWER LEVEL MANAGEMENT AND SUPERVISORY PERSONNEL HAS BEEN LEFT TO, IN LARGE PART, VENDOR COMPANIES. WHAT MANAGEMENT TRAINING THAT DOES EXIST IN ROBOTICS IS PRIMARILY PROVIDED BY VENDORS AS OPPOSED TO INDUSTRY OR MANAGEMENT CONSULTANTS, EDUCATIONAL INSTITUTIONS, OR PROPRIETARY SCHOOLS. THE RANGE OF PROGRAMS IS FROM A 3-WEEK COURSE EMPHASIZING HANDS-ON EXPERIENCE TO INFORMAL WORKSHOPS ON-SITE (CONDON, 1983). TRAINING OF MANAGEMENT BY VENDOR COMPANIES HAS THE DISTINCT ADVANTAGE OF PROVIDING TRAINING AMENABLE TO THE ROBOT SYSTEM PURCHASED, BUT HAS THE POTENTIAL DISADVANTAGES OF BEING VARIABLE IN QUALITY AND NONCOMPREHENSIVE IN NATURE. IN ADDITION, TRAINING BY VENDORS MAY EMPHASIZE ONLY THE POSITIVE ASPECTS OF ROBOT IMPLEMENTATION (SUCH AS INCREASED PRODUCTIVITY) AND MAY DOWNPLAY HUMAN FACTORS SUCH AS WORKER DISPLACEMENT AND WORKER STRESS ASSOCIATED WITH INNOVATION THAT MAY THREATEN EMPLOYMENT. USER ROLES SUCH AS MONITORING AND MAINTENANCE MAY ALSO BE NEGLECTED. THIS INCOMPLETE APPROACH TO TRAINING MAY BE THE RESULT OF INADEQUATE TRAINING STAFF AND EDUCATIONAL MATERIAL PRODUCTION CAPABILITIES, DEMAND FROM CUSTOMERS, AND PERCEIVED ROLE OF USERS IN PROVIDING ADDITIONAL TRAINING. SME (1982) HAS MADE SOME SUGGESTIONS REGARDING POTENTIAL CHANGES THAT WILL AFFECT SUPERVISORY PERSONNEL. PROBABLE TOPICS FOR FUTURE TRAINING EFFORTS INCLUDE INCREASED SKILL REQUIREMENTS AND REORIENTATION, IMPROVED MANUFACTURING CONTROL, SHIFT FROM PERSONNEL TO ADMINISTRATIVE ROLES, AND INCREASED PLANNING AND MAINTENANCE REQUIREMENTS.

MANAGEMENT MAY ALSO EXPERIENCE A NEED FOR IMPROVED HUMAN RESOURCE DEVELOPMENT AND TRAINING WHERE INNOVATION AND HIGH TECHNOLOGY ARE INVOLVED DUE TO THE RISK FACTORS ASSOCIATED WITH PLANNED CHANGES. TRAINING IN GROUP DYNAMICS, STRESS MANAGEMENT, AND BURNOUT MAY BECOME VALUABLE ADDITIONS FOR INTERNAL MANAGEMENT DEVELOPMENT.

NOT ONLY MUST MANAGEMENT BE SENSITIVE TO THE PLANNING FOR AND IMPLEMENTATION OF ROBOTS, IT MUST ALSO MAKE INFORMED DECISIONS REGARDING CHANGE IN THE LABOR FORCE. THE FOLLOWING SECTION DESCRIBES POTENTIAL TRAINING/EDUCATION RELATED PROBLEMS AFFECTING EMPLOYEES DURING THE FACTORY AUTOMATION PROCESS.

Worker Training

BASICALLY, THERE ARE TWO MAJOR SCENARIOS INVOLVING WORKERS FACED WITH JOB AUTOMATION. FIRST, WORKERS MAY BE DISPLACED BECAUSE THEIR JOBS HAVE BEEN ELIMINATED. IN ORDER TO REMAIN IN THE WORK FORCE, LABORERS MUST EITHER UPGRADE THEIR SKILLS THROUGH RETRAINING OR SEEK OTHER EMPLOYMENT. THE SECOND SCENARIO IS THAT AUTOMATION ACTUALLY CREATES JOBS, AND REQUIREMENTS FOR SKILLED TECHNICIANS WILL PROVIDE NEW OPPORTUNITIES FOR EMPLOYMENT.

THE POSITIVE ASPECTS OF UTILIZING ROBOTS IN THE LABOR FORCE ARE THAT THE WORK PLACE ENVIRONMENT SHOULD BECOME SAFER AND CLEANER AS WELL AS LESS HAZARDOUS. THE EFFECTS ON THE WORKER ARE PROJECTED TO BE INCREASED PRODUCTIVITY, MORE ENRICHED JOBS, AND IMPROVED QUALITY OF LIFE, INCLUDING HIGHER WAGES AND MORE LEISURE TIME (SME, 1982).

LOOKING AT THE NEGATIVE ASPECTS OF AUTOMATION AND SINGLING OUT THE IMPLEMENTATION OF ROBOTS, THEOBOLD (1982) PREDICTS THAT THE UNITED STATES IS PROGRESSING TOWARD A PERMANENTLY INFLATED UNEMPLOYMENT RATE AS WELL AS A GENERATION OF UNEMPLOYABLE HIGH SCHOOL AND COLLEGE GRADUATES WHO ARE UNABLE TO FIND JOBS IN HIGH TECHNOLOGY FIELDS. THE MAJOR QUESTIONS THEN BECOME, "WHO IS RESPONSIBLE FOR PROVIDING RETRAINING OF THE NOW UNEMPLOYED WORKER AND IS RETRAINING A SOLUTION TO THE UNEMPLOYMENT PROBLEM?"

IN A RECENT SURVEY OF RETRAINING IN AMERICA, LEE & ZEMKE, (1983) POINT OUT THAT GOVERNMENT HAS MADE FEW SERIOUS EFFORTS IN FUNDING MAJOR JOB RETRAINING PROGRAMS ALTHOUGH SEVERAL BILLS ARE CURRENTLY UNDER CONSIDERATION. MOST OF THE RETRAINING EFFORTS APPEAR TO BE THE RESULT OF CONSORTIUMS OF LOCAL GOVERNMENT, PRIVATE INDUSTRIES, AND

LABOR GROUPS. ALTHOUGH NO SYSTEMATIC EVALUATION OF THESE PUBLIC/PRIVATE INDUSTRY PROGRAMS CURRENTLY EXISTS, WHAT LIMITED DATA THAT ARE AVAILABLE SUGGEST MIXED RESULTS. LEE AND ZEMKE (1983) CITE THE RECENT CALIFORNIA-BASED GM AND FORD/UAW EFFORTS IN THIS AREA. PROGRAMS HAVE BEEN DEVELOPED TO RETRAIN DISPLACED AUTO WORKERS FOR EXISTING JOBS IN THEIR GEOGRAPHIC AREAS. ALTHOUGH OUTPLACEMENT COUNSELING AND RETRAINING ARE GENERALLY VIEWED AS HAVING A POSITIVE EFFECT, ONLY 20 PERCENT OF THE WORKERS ELIGIBLE FOR SUCH TRAINING PARTICIPATE. THE TYPE OF WORKER SEEKING RETRAINING, HOWEVER, MAY BE THE LABORER WHO HAS SKILLS THAT ARE NOT READILY TRANSLATABLE INTO NEW WORK ASSIGNMENTS. THERE MAY ALSO BE AN AGE BIAS AMONG THOSE SEEKING RETRAINING. SOME EVIDENCE SUGGESTS OLDER WORKERS ARE MORE LIKELY TO REMAIN PERMANENTLY UNEMPLOYED THAN TO SEEK RETRAINING. CURRENT ESTIMATES OF A TOTAL OF 63 PERCENT OF WORKERS NEEDING RETRAINING DUE TO ROBOTS ON THE WORK FORCE MAY REQUIRE A CAREFUL ANALYSIS OF THE TYPES OF WORKERS DISPLACED, AGE, CURRENT SKILL LEVEL, ETC., TO RETURN A HIGH PERCENTAGE OF THEM TO THE WORK FORCE.

LOOKING AT THE NEED FOR TRAINING HIGHLY SKILLED TECHNICIANS, A RECENT FORECAST ESTIMATES THAT IN EXCESS OF 400,000 ROBOTICS TECHNICIANS WILL BE NEEDED TO SERVICE AUTOMATED FACTORIES AND ROBOTS BY THE END OF THIS DECADE. GIVEN THAT THERE ARE APPROXIMATELY 10,000 STUDENTS CURRENTLY IN TECHNICIAN TRAINING, A SEVERE PERSONNEL SHORTAGE SHOULD EXIST BY 1990. THIS PREDICAMENT HAS NOT BEEN WASTED ON MANY EDUCATIONAL INSTITUTITONS WHO HAVE DEVELOPED ROBOTICS TRAINING PROGRAMS EMPHASIZING THE NECESSARY HIGH TECHNOLOGY SKILLS NEEDED TO BE COMPETITIVE IN THE INDUSTRY. MOST OF THESE PROGRAMS REPRESENT COOPERATIVE EFFORTS BETWEEN THE INDUSTRY THAT REQUIRES TECHNICAL PERSONNEL AND INSTITUTIONS SEEKING TO UPGRADE THEIR CURRICULA. AVERAGE SALARIES FROM ONE COMMUNITY COLLEGE FOR ROBOT TECHNICIAN POSITIONS ARE IN THE $24,000 PER YEAR RANGE FOR 18 OUT OF 23 GRADUATES WITH ASSOCIATE DEGREES (SCHREIBER, 1981). HOWEVER, ANOTHER PROGRAM WAS ABLE TO PLACE ONLY A FEW OF ITS GRADUATES (LEE & ZEMKE, 1983).

AT LEAST ONE ADVANTAGE OF THE ROBOTICS REVOLUTION HAS NOT BEEN LOST ON TRAINERS. SINCE THE FIELD IS HEAVILY DEPENDENT ON SOFTWARE DEVELOPMENT, CONCURRENT ADVANCES ARE PREDICTED FOR COMPUTER-BASED TRAINING (CBT). WITH THE HIGH LEVELS OF RETRAINING CURRENTLY PROJECTED FOR INDUSTRY AND THE RELATIVELY SHORT-TIME PERIOD ALLOCATED FOR RETRAINING (5 TO 15 DAYS; SME, 1982), CBT IS PREDICTED TO BECOME A MAJOR SOURCE BOTH FOR RETRAINING AND NEW TRAINING. ADVANCES IN HIGH FIDELITY SIMULATION, ARTIFICIAL INTELLIGENCE, COMPUTER VISION, AND INTERACTIVE COMPUTER GRAPHICS SHOULD CONTRIBUTE TO NOT ONLY UPGRADING VENDOR TRAINING PROGRAMS, BUT ALSO IMPROVE THOSE PROGRAMS LAUNCHED IN EDUCATIONAL INSTITUTIONS, SUCH AS VOCATIONAL AND TECHNICAL TRAINING SCHOOLS.

SUMMARY

DESPITE THE FACT THAT THERE ARE LITTLE CONCLUSIVE DATA AND MUCH SPECULATION ABOUT THE IMPACT OF ROBOTICS ON EMPLOYMENT, THE CHANGE IN THE AUTOMATION OF INDUSTRY IS ALREADY UNDER WAY. THOSE INVOLVED IN ADMINISTRATION AND TRAINING, ARE PRESENTLY FACED WITH DETERMINING A REALISTIC ROLE FOR AUTOMATION WITHIN THEIR OWN INDUSTRIES AS WELL AS ASSESSING THE IMPACT OF ROBOTICS ON UNEMPLOYMENT, THE NEED FOR RETRAINING AND REQUIREMENTS FOR SKILLED TECHNICIANS. WITHOUT ADEQUATE FORECAST DATA, OR DIRECTIVES FROM GOVERNMENT SOURCES, IT APPEARS THAT THE PRIVATE SECTOR WITH SOME COMMUNITY AND LABOR SUPPORT WILL BE RESPONSIBLE FOR CREATING MANAGEMENT TRAINING, IMPLEMENTING RETRAINING AND OUTPLACEMENT PROGRAMS FOR DISPLACED WORKERS, AND DEVELOPING EDUCATIONAL PROGRAMS FOR PRODUCING SKILLED LABORERS. THE RATE AT WHICH SUCH PROGRAMS DEVELOP WILL BE INFLUENCED BY NOT ONLY THE RATE OF TECHNOLOGY CHANGE BUT ALSO THE ADOPTION OF NEW TECHNOLOGY THROUGHOUT INDUSTRY. GIVEN THAT ROBOTICS TECHNOLOGY HAS BECOME THE LATEST ADDITION TO THE INDUSTRIAL REVOLUTION, NOW IS THE TIME FOR THE TRAINING COMMUNITY TO BECOME ACTIVELY INVOLVED IN THE DESIGN/DEVELOPMENT/EVALUATION OF AUTOMATION RELEVANT TRAINING PROGRAMS.

REFERENCES

ALBUS, J. S. INDUSTRIAL ROBOT TECHNOLOGY AND PRODUCTIVITY IMPROVEMENT. IN OFFICE OF TECHNOLOGY ASSESSMENT, EXPLORATORY WORKSHOP ON THE SOCIAL IMPACTS OF ROBOTICS. WASHINGTON, DC: CONGRESS OF THE UNITED STATES, OFFICE OF TECHNOLOGY ASSESSMENT, FEBRUARY 1982.

BLUESTONE, B., & HARRISON, B. THE DEINDUSTRIALIZATION OF AMERICA: PLANT CLOSINGS, COMMUNITY ABANDONMENT, AND THE DISMANTLING OF BASIC INDUSTRY. NEW YORK: BASIC BOOKS, 1982.

CLAPP, N. W. MANAGEMENT RESISTANCE TO ROBOTS. IN ROBOTICS TODAY: '82 ANNUAL EDITION. DEARBORN, MI: ROBOTICS INTERNATIONAL, SOCIETY OF MANUFACTURING ENGINEERS, 1982. (ARTICLE ORIGINALLY PUBLISHED IN ROBOTICS TODAY, SPRING 1981)

CONDON, M. STRAIGHT TALK ABOUT ROBOTS. TRAINING AND DEVELOPMENT JOURNAL, 1983, NOVEMBER, 15-24.

HUNT, A., & HUNT, T. HUMAN RESOURCE IMPLICATIONS OF ROBOTICS. KALAMAZOO, MI: W.E. UPJOHN INSTITUTE FOR EMPLOYMENT RESEARCH, 1983.

HUNT, D. INDUSTRIAL ROBOTICS HANDBOOK. NEW YORK: INDUSTRIAL PRESS, INC., 1983.

LEE, C., & ZEMKE, R. RETRAINING AMERICA: SOLUTIONS OR SUGAR PILLS? TRAINING, 1983, MAY, 22-23; 26-30.

OFFICE OF TECHNOLOGY ASSESSMENT. EXPLORATORY WORKSHOP ON THE SOCIAL IMPACTS OF ROBOTICS. WASHINGTON, DC: CONGRESS OF THE UNITED STATES, OFFICE OF TECHNOLOGY ASSESSMENT, FEBRUARY 1982.

ROBOTS: THE PRICE OF PROGRESS. TRAINING, 1983, JUNE, 86.

ROBOTS REVISITED: ONE YEAR LATER (PAUL ARON REPORT #25). IN OFFICE OF TECHNOLOGY ASSESSMENT, EXPLORATORY WORKSHOP ON THE SOCIAL IMPACTS OF ROBOTICS. WASHINGTON, DC: CONGRESS OF THE UNITED STATES, OFFICE OF TECHNOLOGY ASSESSMENT, FEBRUARY 1982.

SCHREIBER, R. R. MEETING THE DEMAND FOR ROBOTICS TECHNICIANS. IN ROBOTICS TODAY: '82 ANNUAL EDITION. DEARBORN, MI: ROBOTICS INTERNATIONAL, SOCIETY OF MANUFACTURING ENGINEERS, 1982. (ARTICLE ORIGINALLY PUBLISHED IN ROBOTICS TODAY, SUMMER 1981)

SOCIETY OF MANUFACTURING ENGINEERS. INDUSTRIAL ROBOTS: A DELPHI FORECAST OF MARKETS AND TECHNOLOGY. DEARBORN, MI: SOCIETY OF MANUFACTURING ENGINEERS, 1982.

STAUFFER, R. N. REACHING FOR TOP MANAGEMENT ATTENTION. IN ROBOTICS TODAY: '82 ANNUAL EDITION. DEARBORN, MI: ROBOTICS INTERNATIONAL, SOCIETY OF MANUFACTURING ENGINEERS, 1982. (ARTICLE ORIGINALLY PUBLISHED IN ROBOTICS TODAY, WINTER 1980-81)

THEOBALD, R. AVOIDING 1984: MOVING TOWARD INTERDEPENDENCE. ATHENS, OH: SWA-LOW PRESS, 1982.

WEEKLEY, T. L. THE UAW SPEAKS OUT ON INDUSTRIAL ROBOTS. IN ROBOTICS TODAY: '82 ANNUAL EDITION. DEARBORN, MI: ROBOTICS INTERNATIONAL, SOCIETY OF MANUFACTURING ENGINEERS, 1982. (ARTICLE ORIGINALLY PUBLISHED IN ROBOTICS TODAY, WINTER 1979-80)

ZEMKE, R. THE ROBOTS ARE COMING! TRAINING TOMORROW'S HIGH-TECH WORKERS, TRAINING, 1983, JUNE, 18-20; 26-27; 29-30.

FEDERAL AVIATION ADMINISTRATION

COMPUTER BASED INSTRUCTION--YEAR 10

BARBARA J. WRIGHT

JOHN A. BUCK

ABSTRACT

In October 1982 the Federal Aviation Administration (FAA) Administrator approved the use of Computer Based Instruction for the third major technical occupation in FAA, Air Traffic Controllers. The first terminals were in place to deliver proficiency and refresher training at Air Traffic field locations by the end of 1983.

In September 1983, an expansion of the current CBI training program to engineers and electronics technicians was approved, nearly doubling the number of field sites currently delivering CBI to that occupation.

The implementation of CBI in a large distributed system has numerous advantages including flexible arrangements for training at operational work locations, standardization of training delivery, individualization of training based on employee needs, reductions of travel and per diem costs, instructor time and student or employee time savings. However, there are also a considerable number of program implementation and administrative challenges not the least of which include the need for modifications in attitudes as well as training procedures on the part of students, instructors and program administrators. Participation and involvement in the change process on the part of all of those affected does, we believe, help to assure a more positive acceptance and reduce potential problems and constraints

This article summarizes the FAA's current program, accomplishments to date and some of the ongoing program challenges.

BACKGROUND

The FAA employees about 45,000 persons most of whom work in three major technical occupations including aviation safety, engineering and electronics maintenance and air traffic control. The major portion of FAA's technical training has been provided at a centralized training facility in Oklahoma City, the FAA Academy. The Academy employees about 600 instructors, 500 of which are drawn from the Agency's workforce. Others are obtained via contract during periods of high training demand.

Definitions - CBI Uses

The FAA has used Computer Based Instruction (CBI) as the umbrella term which encompasses computer assisted instruction (CAI) and computer managed instruction (CMI).

Computer Assisted Instruction

In CAI the computer actually delivers instruction generally in one of three modes: drill and practice, tutorial or simulation. Tutorial instruction generally involves basic presentation of information interspersed with questions. For its purposes FAA has found this method most effective where it is also enhanced with graphic simulation. 1/ The FAA uses a considerable amount of sophisticated simulation of actual electronic equipment parts and processes, radar control, ground navigation equipment, flight decks for ground school pilot training, takeoff, landing, and other airplane procedures for air traffic control. Simulation is used to cause the student to replicate all or

1/ The views expressed are those of the authors and do not necessarily reflect the views of the FAA.

part of an actual work process.

Computer Managed Instruction

CMI refers to the built in system for managing training, including test results, training prescriptions, student progress, analysis of information and training evaluation.

CBI differs from nonautomated individualizeded self-paced instruction in that it is "interactive". The student must respond to questions or perform in job-like functions or situations pushing appropriate buttons or levers which bring about an immediate response (learning reinforcement) from the computer. The student advances to successive levels of learning difficulty only as s(he) completes each prerequisite level. The student cannot be passive; s(he) is actively engaged in the learning process.

CURRENT DIRECTION

FAA's current direction with computer based instruction is based on the conclusions of a comprehensive feasibility study that was completed in 1979.[1] The study identified a number of courses with a high student population and a high cost. It compared the potential benefits of several computerized and noncomputerized alternatives for their delivery. It concluded that a geographically distributed system was the most desirable option from the standpoint of cost savings, such as travel, per diem and less quantifiable benefits such as travel, per diem ans less quantifiable benefits such as improved standardization and job site scheduling flexibility. Although cost payback continues to be an important consideration, FAA's experience with CBI to date has confirmed the almost equal importance of the less quantifiable features; training standardization, easily available refresher and update training, more flexible student scheduling, more rigorous course development, improved training performance records, less student time required to complete training and equal and better retention of course information.

Following the comprehensive feasibility study the Interim System, the first large scale field test, was proposed and approved in 1979. CBI training was delivered to engineers and electronics technicians in 65 FAA field locations and nationwide. Student results as measured by standard tests demonstrated that CBI was as effective and often more effective than conventional classroom training. Another measure of student performance compared student results in terms of their success in subsequent courses for which the CBI course provided an essential prerequisite. Again it was demonstrated that CBI students performed at least as well as students who had taken conventional training and often better.

CURRENT PROGRAMS

Coursewares

A total of 14 CBI courses are currently available for the first technical occupation in FAA to obtain CBI, the Flight Inspectors in the Aviation Safety Program. These courses are delivered primarily in a CMI mode. Although computer controlled, this delivery method includes the use of a number of other individualized learning resources, audiovisual equipment, student workbooks, etc. In 1983, thirteen CBI courses were available, on line, for the second major technical occupation in FAA, engineers and electronic technicians. Another fourteen courses are expected in 1984. These courses are managed in a CMI mode and incude from 10 to 25% CAI. For Air Traffic Controllers the third technical occupation to obtain CBI, about 100 lessons are expected by early 1985 for enroute facilities[2] and another 100 lessons for terminal field facilities. These lessons are designed primarily in a CAI mode. Only the student testing, evaluation and record keeping are in the CMI mode. The initial air traffic lessons will be used as part of the structured field program of proficiency and refresher training. New controllers will also begin to receive some training during 1984 in a CBI mode.

System Configuration: Evolution Toward Portability

Experimental work with CAI began in 1974 at the FAA Academy. FAA conducted small scale field tests of CBI in 1977. At that time CMI capabilities supported by mainframe services became available for management of training at remote locations. At the time of this writing, software and telecommunications were provided via a mainframe computer over a prescribed telecommunication network at a site removed from the FAA Academy for the majority of the existing FAA/CBI courses. FAA recently procured a new

1/The FAA Feasibility Study and the History of the FAA CBI Program is documented in previous ADCIS Articles by John Buck in 1979, 1980 and most recently in November 1981, "CBI in FAA--Year 7."

2/En route facilities provide air traffic separation and other services to en route aircraft; terminal facilities provide similar services for departing and arrival aircraft within the confines of the terminal airspace.

generation of learning terminals whose specifications were designed at the National Library of Medicine. These terminals accommodate FAA's CBI lessons using the current authoring languages as well as other languages such as PASCAL, PILOT and FORTRAN 77.

The new terminals also differ in that they accommodate standard (ASCII)[1] telecommunications protocol which, for the first time, permits FAA to have the flexibility to connect to a greater number of telecommunications networks. The FAA CBI staff continually examines the cost benefit ratio of various telecommunications and software service options. The availability of standard telecommunications could ultimately result in considerable reductions in software and telecommunications costs. An on-line mainframe system is still required to accomodate the complexity of current lessons designed for aviation safety and electronics maintenance training and the software requirement for comprehensive training management of those programs. In addition, it is more efficient to design the more complex simulations required by those programs using the current "on-line" authoring system. However because of ASCII, there are now a number of alternative ways to reducing the cost of that service which are currently being evaluated. FAA has been using a different authoring language to design and deliver CBI to air traffic controllers in an "off-line" mode. "Off-line" and "stand-alone" are used synonomously in FAA to connote the delivery of CBI without telecommunications support using a self-contained micro-computer and associated disks for local storage of software and courseware. Although this provides substantial reductions in telecommunications expenses, other problems are created. The available systems for CMI in an "off-line" mode have not had the sophistication of the on line CMI system currently used by FAA. The air traffic lessons are delivered via a self-contained micro-computer embedded in the learning terminal and for the present use floppy disks and accompanying disk drives for software storage. FAA presently envisions an 80/20 distribution of on-line vs off-line time. That is, the lessons will be delivered off line which will consume 80% of the CBI time while student tests, lesson validation and other training imformation data will be delivered over telephone lines back to a mainframe computer for centralized program quality control and management.

The other FAA CBI programs will also use disk drives and floppy disks to enhance some of their CBI lessons in 1984.

Stand-alone or off-line lesson delivery has significant advantages for complex simulations in that the screen plots at a much faster rate in a stand-alone mode saving student time and telecommunications time. FAA is currently involved in ongoing studies to determine the most efficient mix of on-line vs. off-line delivery for all of its CBI programs.

PROBLEMS AND CHALLENGES

Stand-Alone Course Management

One major disadvantage of off-line course delivery in 1983 was the unavailability of a management system as sophisticated as the FAA's on-line system for the management of training information delivered in a stand-alone mode. In 1983 FAA completed a study of the available options for managing stand-alone or portable CBI and determined that workable alternatives did exist but will probably have to be modified. This study also examined the benefits of a hard disk vs. floppy disk configuration for the storage of training inforamtion to be delivered in a stand-alone mode. FAA opted to use a floppy disk configuration in the near term but may move to a hard disk configuration the future as the technology evolves.

Most Efficient Local Site Administration

FAA is involved in a determination of the most effective means of decentralized site administration for CBI. Field sites have been experimenting with a variety of administration procedures since 1981. This inforamtion is currently being analyzed and will form the basis for the issuance of standard procedures.

Quality CBI Orientation; Effective Management Support

Local site administration procedures will differ for CBI systems that run in on line vs. stand-alone mode. However, there are common denominators. It is clear that the extent of local management support[2] and the quality of the initial student orientation affects students' attitudes toward CBI as well as initial progress through the course. the number of questions asked by students is reduced by adequate orientation. However, the reasons for variation in the amount of questions asked by students are currently being studied. These reasons may include: a student's understanding of how to study most efficiently in an individual mode, the student's familiarization with prerequisite informaiton or the extent of CAI or interactivity in the CBI course.

1/ American Standard Code for Information Interchange; Accepted as Standard by the Industry.

2/ Overcoming organizational resistence to CBI is covered in an ADCIS article published in the 1983 Conference Proceedings called: "How to Answer the Question: What's A CBI?" by John Buck and Barbara Wright.

CBI Lesson Design & Development

There is a substantial amount of time required for those inexperienced in CBI course design to learn to create high quality individualized courseware. Therefore substantial revisions were needed for some of the early courses. The results of student evaluation during the prototype phase of course development have substantiated qualitative improvements in courseware, including the addition of more CAI. Course development procedures based on the requirements for development of classroom courses have been changed substantially for CBI. Although field personnel can still be detailed to work in course development as they have in conventional course development, a permanent cadre of highly trained CBI developers and programmers are also required. CBI development is considerably more complex and more time consuming then conventional classroom development. CAI requires from 100 to 300 hours of development time per course hour depending on the experience of the developer and the complexity of the simulation.

Course Development Standards

Although FAA has made considerable strides in this area substantial challenges remain. FAA participates in a consortium of Federal agencies working in the CBI area, the Federal Interagency Group for Computer Based Training (IGCBT). The group has as a primary objective in 1983 and 1984, the design and promulgation of CBI standards that could be used for both in house course development and obtaining quality products from the vendor community. The need has been acknowledged and addressed in a cooperative way by both groups.[1]

Degree of Work Task Simulation

Additional attention now needs to be given to the question of how much on the job training particularly hands on or laboratory equipment training can effectively be delivered in a CAI format. To respond to this question FAA will continue to explore the potential of tools such as video disk, compressed sound, artificial intelligence and adaptive testing.

Timely Adoption ofBeneficial System Enhancements

Another challenge is keeping abreast of advancing technology and determining when it is most efficient in terms of both learning and cost effectiveness to replace or adapt existing technologies. Continual analysis and communication with other users is important. FAA stays in communicaiton with government agencies, the airlines, other industries and professional organizations both in the U.S. and abroad.

Portability

Many new questions have had to be addressed in the process of implementing a more portable CBI system. However, there is little doubt that portability is a viable objective as long as sufficient attention is given to the security and preservation of test results, evaluation and automated collection of appropriate other training management information in a timely manner.

In 1983 one of FAA's local field sites initiated an experiment in enhanced portability with the use of mobile van to bring CBI to more remote locations. To negate local telecommunications costs, Remote Microwave (RML) links are being used to link the on-line mobile unite to the nearest telecommunications line. That experiment is still in progress.

SUMMARY

FAA has experienced considerable success in the use of CBI in terms of learning effectiveness, the responsiveness of CBI to the agency's training needs at remote work locations and cost effectiveness. The program is also helping FAA to accomplish high priority human relations initiatives by providing training locally enabling employees to reduce their travel and accompanying period of family separation.

However, there have been and may continue to be many "growing pains"[2] as trainers, managers and local supervisors change attitudes and adopt new procedures to optimize the efficiency of the new system.

1/Barbara Wright has worked with John Buck managing instructional technology systems in FAA since 1982. prior to that she managed a computer based training pilot program at the Social Security Administration.

2/To reduce the "growing pains" of organizations just beginning in CBI the Federal IGCBT published a Starter Kit. The kit can be obtained from the National Technical Information Service. Ask for Report No. FIGGBT/DOC-1 Computer Based Training Starter Kit NTIS Access-No. PB-83-248815.

AUTOMATED INSTRUCTIONAL MANAGEMENT

AIM/3000

John Ackelson, School District No. 12, Adams County, Colorado

Dr. Ted Rogers, School District No. 12, Adams County, Colorado

ABSTRACT

The AIM/3000 system was designed in order to facilitate the management of instruction at the classroom level in a mastery learning environment and to allow aggregation of data for use at various levels within the organizational structure. The system was further designed to interface with other major systems within the overall management information system. The system components include: (1) instructional objectives data bank; (2) instructional resource system; (3) test item bank; (4) student diagnostic capabilities; (5) student progress reporting system; and (6) aggregate of data at defined levels. The perceived benefits of the system for instructors include: (1) reduction in time for record keeping activities; (2) reduction in time in locating resources needed for instruction; (3) generation of exams; (4) scoring of exams; and (5) providing data for managing classrooms in terms of levels of achievement by students. The system is being implemented in six beta sites which includes two elementaries, two junior highs and two high schools. Data gathered from the beta sites will be evaluated in terms of instructor's time necessary to operate the system, utilization of the system and influence on student achievement.

School District No. 12, Adams County began the development of instructional objectives and methods of measuring the achievement of students in relationship to these objectives in 1973. Since then, the district has developed and refined instructional objectives in the curricular areas of mathematics, reading, language arts, social studies, science, foreign language, music, art, health and physical education.

The measurement techniques associated with each objective have been developed and examined for validity and reliability. As the objectives are reviewed and further refined, so are the measurement methodologies.

As instructors taught to the instructional objectives, several problems emerged in terms of management or instruction. Some of these problems included:

1. Record-keeping of student achievement of the instructional objectives.
2. Development of reliable and valid exams for formative evaluation purposes.
3. Scoring of examinations.
4. Obtaining diagnostic information concerning student achievement in regard to process (thinking skills) and content (subject matter).
5. Correlating instructional objectives from one curricular area to the other, so redundant instruction did not occur.
6. Reporting to parents concerning mastery of objectives for students.

BEST COPY AVAILABLE

7. Locating resource material for teaching of the instructional objectives and a description of the resource material, in terms of its relation, to the objectives.

The solution in regard to these problems, was an automated system. However, for a system to be able to address the described problems, there were a number of parameters that had to be considered. These parameters included:

1. The system had to be able to accommodate the data requirements of 1,000+ instructors and 19,000+ students in various curricular areas.
2. The system had to have the capacity of maintaining records in the ten curricular areas for the student's entire stay in the district, be it for 1 month or thirteen years, and be capable of being accessed by instructors who need to examine the data.
3. The system had to have the capability of generating various aggregations of data, based on the organizational structure, and its need for data. Therefore, the system had to aggregate data at the (a) classroom level, (b) grade within a school level, and (c) grade within the district level.
4. The system had to be very user friendly.
5. The system had to be integrated into the MIS and be interfaced with the student information system, summative testing system, library system, attendance system, grading system, and financial system.

As the district began to address the problems and the above defined parameters, a number of existing systems were examined by a team of instructors, administrators and system analysts. This process began in 1980 and ended in October of 1982.

The major CAI/CMI systems were examined, in detail, as well as a number of micro-systems. The major conclusions of this 2 1/2 year study were:

1. The examined systems were not compatible with the district MIS system unless extensive and complex modifications were made.
2. The examined systems were designed for training environments and not a public education environment.
3. There were no provisions for mass data input.
4. No provisions were made for aggregation of data beyond the classroom.
5. The hardware necessary to drive the systems was very expensive.
6. The examined systems could not address the district library-media resources and the necessary information needed for each.

With these conclusions, it was decided to develop a CMI system for use in the public education environment. Financial support for the development of the system came not only from district funds, but from Hewlett-Packard and, to a lesser degree, National Computer Systems.

The system was designed with input from over 130 people, including classroom instructors, school administrators, curriculum specialists, media specialists, district administrators, school board members, system analysts and hardware and software vendors.

The system is currently being tested at six beta sites consisting of two elementaries, two junior high schools and two high schools. Data from the beta sites will be available in April.

Data analysis will focus on three elements. These include: (1) instructor time necessary to operate the system, (2) utilization of the system by the instructors, and (3) student achievement.

OVERVIEW OF AIM/3000

The use and operation of AIM/3000 is centered around a number of system components. The primary focus of AIM/3000 is directed at instructors and their management of instruction.

The major components of the CMI System include:

1. Instructional Objectives Data Bank,
2. Instructional Resources Data Bank,
3. Test Items Data Bank,
4. Student Diagnostic Capabilities,
5. Student Data Bank/Student Progress Reporting. and
6. Aggregation of Data at Various Levels.

INSTRUCTIONAL OBJECTIVES

For each instructional objective in the system the following information is displayed:

1. A brief description of the objective,
2. A complete description of the objective,
3. A taxonomical classification for the objectives,
4. Skill level,
5. Difficulty level,
6. Mastery criteria,
7. Correlated instructional objectives in the same or other curricular areas, and
8. Number of items in the item bank for the objective (this element is system generated).

RESOURCE DATA BANK

The resource data bank allows instructors to locate instructional material, based upon a particular instructional objective.

When an instructor wishes to locate resources, the CMI System prompts the instructor for the short code of the desired instructional objective. The objective description is retrieved and displayed, along with the available resources and their learning modes and locations within the school district.

TEST ITEM DATA BANK

Instructors will utilize the TEST ITEM DATA BANK to generate tests to measure the mastery level of their students, in relation to specific instructional objectives. The AIM/3000 System will prompt an instructor to identify the instructional objectives to be tested, the number of test items to be included for each objective, the types of questions to be included (TRUE/FALSE, MULTIPLE CHOICE, etc.) and the range of the difficulty of the items. To aid the instructor in assessing the difficulty level of the items, a continuous record is kept that displays the number of responses for each foil for each item. The CMI System selects items from the TEST ITEM DATA BANK that meet the instructor's criteria, and transmits them to the printer at the instructor's school, along with the appropriate key.

In addition to the system generating the exam for the instructors, three other methods for test definition are available. The first method is a manual selection of each item as desired on the exam by the instructor. The instructor simply enters into the item bank and selects the exact item they desire for the exam. The second method is that an instructor can develop a teacher-made exam and then define the exam in terms of the objectives tested and the relationship of the items to the objectives within the system. The system will then score the exam, update the student records as appropriate and generate reports from the testing, as if it was a system-generated exam. The third alternative is to use a previously defined exam in the test library of the system.

Students respond to the question on Optical Mark Sense scan sheets, which are transmitted back to the central computer for updating of the CMI Data Base. In addition to updating the CMI Data Base for each student, diagnostic and item analysis reports are produced and transmitted back to the instructor's school for on-site printing. The student diagnostic report is capable of providing process, as well as content data, back to the instructor.

Three Test Analysis Reports are produced with the scoring of each CMI test:

1. Frequency Distribution Report,
2. Student Mastery Item Analysis Report, and
3. Student Diagnostic Report.

STUDENT DATA BANK

The AIM/3000 maintains basic information about each student in the district. This includes for example, student name, grade level, sex, school I.D., enrollment date, home room number, and the various courses that a student is presently enrolled. In addition, the AIM/3000 System maintains instructional mastery information for all students. The system maintains for the current year, the instructional objectives that each student has mastered, including the date of this mastery. The system also maintains mastery history for each student, identifying by sub-curriculum areas (e.g., Reading-Grade 3), those instructional objectives that a student has mastered during prior years.

STUDENT PROGRESS REPORTING

The mode of data input from the various schools for updating student mastery of objectives varies. In most cases, data entry occurs through an Optical Mark Reader located at the school site. In some cases, the data entry is on-line through a CRT.

The type of mastery data transmitted through a school's Optical Mark Reader generally is student test data. Teacher observation data is also sent to the central computer; some objectives are not measurable by pencil and paper; however, an instructor may be able to observe behavior to determine mastery. These observations still come into the system as mastery of objectives and update student mastery; however, no scoring is necessary as is the case with student test data.

In addition to the on-line displays for student progress, four Student Progress Reports are available through the CMI System:

1. Student Mastery Report,
2. Class Mastery Report,
3. School and District Mastery Report by Grade, and
4. Percentage of Students Mastery Report, Total District.

CONCLUSION

The AIM/3000 System was designed to meet the needs of instructors who are engaged in teaching to defined instructional objectives and are involved with on-going assessment of student achievement concerning mastery of the instructional objectives. It was also designed as part of an overall education MIS, which would allow decision makers, at various levels within the organization, to access pertinent and timely data.

The system should reduce significantly the instructor's time needed to perform clerical tasks, in relationship to the district's mastery learning system. This will be determined by April, as beta site testing is completed.

In brief, the perceived benefits to the instructors include:

1. Reduction of time in locating resources needed for instruction,
2. Generation of exams,
3. Scoring of exams, and
4. Providing data for managing their classroom in terms of achievement.

COMPUTER LITERACY: IS IT IMPROVED BY THE USE OF MICROCOMPUTERS IN CONTENT COURSES?

MARIANNE J. D'ONOFRIO - UTAH STATE UNIVERSITY
JANE D. NATE - KEARNS HIGH SCHOOL, UTAH

MARK E. SLAMA - UTAH STATE UNIVERSITY

ABSTRACT

This study assessed the impact of microcomputer use on the computer literacy of high school accounting students. The non-equivalent control group design was used to test the hypothesis that a significant difference in computer literacy would result between students exposed to microcomputers and students not exposed in a non computer science course. A significant difference in computer literacy was not found to exist between the experimental and control groups. Results were compared with a similar study which found gains in computer literacy to be associated with exposure to microcomputers in a non computer science course. Inferences were drawn as to reasons for the conflicting results.

INTRODUCTION

The use of microcomputers has grown substantially during the past few years and is having a considerable impact on society (Beck, 1980). In businesses alone it is estimated that almost 750,000 persons are currently using microcomputers and their use is expected to expand at tremendous rates (Uthe, 1982; Kearsley, Hillelsohn & Sudel, 1981). This increased usage of microcomputers makes computer skills especially important to this generation of business students. In fact, Michael (1968) suggests that ignorance of computers will soon render people functionally illiterate as does ignorance of reading, writing, and arithmetic.

A term used to describe proficiency with computers is computer literacy (Klassen, Anderson, Hansen, & Johnson, 1980). One method of increasing students' computer literacy is to teach them directly about the computer in computer oriented courses. A less direct but perhaps supplementary method is for students to become computer literate through the use of computers in traditional courses such as accounting. This study deals with the impact of computer assisted instruction in accounting on computer literacy.

Computer Literacy

The term computer literacy made its way into academic vocabulary during the past decade and has gained much publicity in the last few years. At present, there is not a firm definition of computer literacy. Nevertheless, computer literacy continues to evolve as educators and lay persons alike struggle to sort out what computer literacy includes (Masat, 1981). Various authors (Johnson, Anderson, Hansen, & Klassen, 1980; Klassen et al., 1980) have described computer literacy as: (1) imparting knowledge about handling information; (2) dispelling fears and myths associated with computers; (3) developing skills in using and programming a computer; (4) developing procedural learning; and (5) addressing the ethical and societal issues raised by computing.

Computer-Assisted Instruction (CAI)

The uniqueness of the computer as an interactive and responsive tool makes it an effective educational aid in instructional systems. In education, the computer provides instruction in the form of drill-and-practice, tutorials, individualized instruction, simulation, problem solving, and testing. CAI can also be particularly useful in basic skills courses, computer science, mathematics, and computer literacy courses.

Most research in the area of CAI has investigated the impact of CAI on student achievement in a variety of content areas. Many of these studies indicate that when CAI is used as a supplement to traditional instruction, student performance is consistently superior to that from traditional instruction alone. When used as a substitute for traditional teaching methods, the computer approach results in comparable or superior performance in comparison to other strategies (Bray, 1981).

Importance of Study

The present study was undertaken to increase knowledge about computer literacy resulting from the use of microcomputers in computer-assisted instruction. This study replicates a study completed by D'Onofrio and Slama (1983, in press). Replication of research is viewed by contemporary leaders in educational research (Best, 1970; Borg & Gall, 1979; Shaver & Norton, 1980) as a strategy for establishing the generalizability of findings or determining the conditions under which the results will be supported. Thus, the present study was undertaken repeating the D'Onofrio and Slama study under different conditions of time and place.

STATEMENT OF PROBLEM AND METHODOLOGY

This study investigated the extent to which computer literacy resulted from the use of microcomputers in courses unrelated to computer science. The two accounting classes at the Western high school participating in the study were intact groups. Thus, the quasi-experimental (non-equivalent control group) design was employed in this study (Campbell & Stanley, 1963). One accounting class used microcomputers to solve accounting problems, while the other class solved accounting problems using traditional methods. The treatment (using microcomputers) was randomly assigned to one of the two classes. The control group had 21 students, one-third of the students male (Table 1). Twenty-six students were in the experimental group, with slightly more than one-third being male.

The cognitive and affective computer literacy of the two groups were assessed one week before the groups began their accounting units, designed to reinforce the accounting cycle. After six weeks, the cognitive and affective computer literacy of the two groups were again assessed. The two groups participating in the study had completed the study of the complete accounting cycle and had demonstrated their knowledge of the accounting cycle using the manual (traditional) method of accounting prior to the commencement of the study. The computer-assisted accounting unit (Allen & Klooster, 1982) used by the experimental groups is designed to introduce students to computerized accounting principles and reinforce the accounting students' understanding of the accounting cycle. The computer-assisted instruction accounting unit consists of four major integrated accounting systems: General Ledger, Accounts Payable, Accounts Receivable, and Payroll. The experimental group completed the first three accounting systems. The control group completed a manual accounting practice set (Swanson, Ross, Hanson, & Boynton, 1980), designed to reinforce the students' understanding of the accounting cycle through a simulation of the accounting cycle of a particular firm.

Instrument

The Minnesota Computer Literacy and Awareness Assessment questionnaire, Form 8 (Anderson, Klassen, Hansen, & Johnson, 1979) was the instrument used to assess computer literacy of the high school accounting students. The instrument was developed to reflect a broad conception of computer literacy including several cognitive and affective domains. More specifically, the following components of affective computer literacy were assessed: computer anxiety, computer efficacy, and computer enjoyment. The components of cognitive computer literacy that were measured included: applications, capabilities, programming, and software.

The affective dimensions included required students to respond to 15 items. Item alternatives were strongly disagree, disagree, undecided, agree, or strongly agree. The affective items are designed to assess the attitudinal components of computer literacy. The cognitive assessment consists of 28 items containing either true-false, or multiple choice items and all contain an "I don't know" option (Anderson, Krohn, & Sandman, 1980, p. 3).

Verification of the a priori dimensions of affective computer literacy was accomplished by performing a factor analysis (Anderson et al., 1980). Three factors corresponding to the affective computer literacy subscales used in this study resulted and are described below (Anderson et al., 1980, p. 5):

Anxiety. The level of stress

that is associated with dealing with computers.

Efficacy. The extent to which a student feels confident about his or her ability to deal with computers.

Enjoyment. The degree to which a student enjoys computers or learning about computers.

The alpha reliabilities (Anderson et al., 1980) and Spearman Brown corrected alphas for the three affective subscales are given in Table 2.

The dimensionality of cognitive computer literacy was assessed (Anderson et al., 1980) using factor analysis. This resulted in the four factors (which correspond to the cognitive subscales) described below.

Applications. Applications of computers in society.

Capabilities (Limitations). The tendency to attribute capabilities to a computer that it does not possess.

Software. Simple vocabulary items such as the definition of a program as well as some advanced items requiring an understanding of algorithmic problem solving.

Programming. Ability to read a BASIC program and understand basic concepts such as multiple replacement, conditional iteration, and program modification.

The alpha reliabilities (Anderson et al., 1980) and the Spearman Brown corrected alphas for the four subscales (Table 3) indicate reasonable levels of internal consistency considering the number of items included in each subscale.

ANALYSIS

The major purpose of the data analysis was to determine whether the use of microcomputers in a non-computer science class increased computer literacy. Changes in the scores on the subscales of the Minnesota Computer Literacy and Awareness Instrument, Form 8, were compared between the experimental and control groups (Campbell & Stanley, 1963). Raw change score analysis rather than analysis of covariance was appropriate given that selection into experimental and control groups was based on intact classes with pre- and post test variances not significantly different.

The statistical technique used was Multivariate Analysis of Variance (MANOVA). MANOVA was used to assess whether differences in the categorical independent variable were associated with differences in the multiple intervally scaled dependent variables. MANOVA assumes interval scaled dependent variables. The summative attitude scales used in this research (as opposed to one item scales) are generally assumed in the social sciences to provide interval scale data (Churchill 1979; Green 1978) and this assumption leads to conservative estimates of statistical relationships (Bohrnstedt, 1970).

MANOVA was used to determine if differences existed between the experimental and control groups with regard to their change scores on the computer literacy subscales. In particular, MANOVA was used to test the following hypothesis:

H_0: There are no significant differences in the changes in computer literacy between the group exposed to the microcomputers and the group not exposed.

The dependent variable computer literacy was operationalized by computing change scores for the following computer literacy subscales: computer anxiety, computer applications, computer efficacy, computer software, computer capability, computer enjoyment and computer programming.

The research hypothesis could not be rejected ($p = .14$; Table 4). Therefore, no statistically significant difference was found between the vectors of change scores for the experimental and control groups. However, some aspects of computer literacy seemed to change more than others with exposure to microcomputers. The subscale scores which tended to vary the most with exposure to microcomputers concerned computer applications, computer capabilities, and computer software (Table 4).

SUMMARY AND DISCUSSION

The results of this study could not support the notion that the computer literacy of high school students would increase more with exposure to microcomputers in a content related course than without exposure to microcomputers. However, in the D'Onofrio and Slama (1983, in press) study, students exposed to microcomputers while learning accounting

increased their computer literacy significantly more than the control group. The difference in the results of the two studies may be explained by the different conditions under which the two studies were conducted.

The present study was conducted over a 6-week period whereas the D'Onofrio and Slama (1983, in press) study covered an 8-week period. Thus, those students using the microcomputers were exposed to them for a shorter period of time. In addition, the students only had the microcomputers available to them during their class time. In the D'Onofrio and Slama (1983, in press) study, the students using microcomputers had them available for their use during class time and free periods and before and after the school day; in addition, students were scheduled for microcomputer use two 50-minute periods each week to facilitate completion of their assignments. The students using microcomputers also had their own workbooks and software (disc). Students in the present study had their own workbooks but every two students shared a disc.

In summarizing the results of the two studies, it would appear that some influence on computer literacy results from exposure to microcomputers in content related courses but that the influence becomes more significant with the length of exposure and the amount of out of class computer activity experienced by the students.

REFERENCES

Allen, W. W. & Klooster, D. H. Automated accounting for the microcomputer. Cincinnati: South-Western Publishing Co., 1982.

Anderson, R. E., Klassen, D. L. Hansen, T. P., & Johnson, D. C. Minnesota computer literacy and awareness instrument, form 8, St. Paul, Mn: Minnesota Educational Computing Consortium, 1979.

Anderson, R. E., Krohn, K., & Sandman, R. User guide for the Minnesota computer literacy and awareness assessment. St. Paul, Mn: Minnesota Educational Computing Consortium, 1980.

Beck, J. J. Jr. Computer literacy for elementary and secondary teachers. Paper presented at the annual meeting of the Texas Association for Supervision and Curriculum Development, Houston, November 1980.

Best, J. W. Research in education (2nd ed.). Englewood Cliffs, Nj: Prentice-Hall, 1970.

Bohrnstedt, G. W. Reliability and validity assessment in attitude measurement. G. F. Summers (Ed.), Attitude measurement. Chicago: Rand McNally, 1970.

Borg, W. & Gall, M. Educational research, New York: Longman, 1979.

Bray, R. M. Evaluation of the TIPS computer management system in a social psychology course. Teaching of Psychology, 1981, 8(3), 139-142.

Campbell, D. & Stanley, J. Experimental and quasi-experimental designs for research. Chicago: Rand McNally, 1963.

Chambers, J. A. & Bork, A. Computer-assisted learning in U.S. secondary/elementary schools (Research Report No. 80-03) Fresno, Ca: California State University, Center for Information Processing, 1980.

Churchill, G. A. Marketing research: Methodological foundations. Hinsdale: The Dryden Press, 1979.

D'Onofrio, M. J. & Slama, M. E. An investigation of the computer literacy resulting from the use of microcomputers in high school accounting classes. Manuscript submitted for publication, 1983.

Green, P. E. Analyzing multivariate data. Hinsdale: Dryden Press, 1978.

Johnson, D., Anderson, R. E. Hansen, T. P., & Klassen, D. L. Computer literacy: What is it? Mathematics Teacher, 1980, 73(2), 91-96.

Kearsley, G., Hillelsohn, M. J., & Sudel, R. J. The use of microcomputers for training business and industry. Alexandria, Va: Human Resources Research Organization, 1981.

Klassen, D. L., Anderson, R. E. Hansen, T. P., & Johnson, D. C. A study of computer use and literacy in science education. Final report, 1978-1980. St. Paul, Mn: Minnesota Educational Computing Consortium, 1980.

Masat, F. E. Computer literacy in Higher education. Washington D.C.: Clearinghouse on Higher Education, 1981. (ERIC Document Reproduction Service No. ED 214 446)

Michael, D. N., The unprepared society. New York: New York Basic Books, Inc., 1968.

Molnar, A. R. The next great crisis in American Education: Computer literacy. EDUCOM Bulletin, 1979, 14, 2-6.

Petruk, M. W. Microcomputers in Alberta schools. Alberta, Canada: Edmonton Planning and Research Branch, 1981. (ERIC Document Reproduction Service No. ED 205 214)

Poole, L. H. Computer literacy: The new mandate for general education in the 80's. Saranac Lake, Ny: North County Community College, 1982. (ERIC Document Reproduction Service No. ED 214 610)

Shaver, J. P. & Norton, R. S. Randomness and replication in ten years of the American Educational Research Journal. Educational Researcher, 1980, 9, 9-15.

Swanson, R. N., Ross, L. D., Hanson, R. D., & Boynton, L D. Cycle center. Cincinnati: South-Western Publishing, 1980.

Uthe, E. F. The computers in our lives. Voced, 1982, 57(3), 26-29.

Table 1

Characteristics of Secondary School Accounting I Students Participating in Microcomputer Study

	Number of Cases Control Group grade level				Number of Cases Experimental Cases grade level				Grand Total
Variable	10	11	12	Total	10	11	12	Total	
Sex:									
Male	0	2	5	7	0	4	6	10	17
Female	2	6	6	14	1	5	10	16	30
Total	2	8	11	21	1	9	16	26	47

Table 2

Reliabilities of the Affective Subscales

Scale	No. of Items	Alpha Reliability	Spearman-Brown Prophecy for 40 Items
Enjoyment	5	.81	.93
Anxiety	5	.64	.92
Efficacy	5	.60	.97

Table 3

Reliabilities of Cognitive Subtests

Subtest	No. of Items	Alpha Reliability	Spearman-Brown Prophecy for 40 Items
Applications	12	.68	.88
Software	7	.58	.89
Capabilities	5	.59	.92
Programming	4	.39	.87

Table 4

The Influence of Exposure to Microcomputers on Computer Literacy

	Univariate F	Significance of F	Wilks Lambda	Significance of Lambda
All Variables			0.765	0.147
Anxiety	0.393	0.534		
Application	3.930	0.054		
Efficacy	0.050	0.824		
Software	4.696	0.036		
Capability	4.514	0.039		
Enjoyment	0.006	0.941		
Programming	0.763	0.387		

READING COMPREHENSION SKILLS AND COMPUTER LITERACY

Harriet Johnson, Ed.D.

Hunter College, CUNY

ABSTRACT

This paper considers the importance of reading comprehension skills for computer literacy. Studies of aptitudes involved in computer courses demonstrate that reading comprehension, among other variables, is as reliable a predictor of computer course performance as programmer aptitude tests. Assessments of a college computer science department chairman and of students enrolled in both computer courses and college reading courses, and examination of computer assignments and documentation confirm the relationship between reading comprehension and computer use. These studies and assessments suggest that college reading courses have an added significance for basic skills students who want to become computer literate.

Statistics on increasing computer use are overwhelming: Guthrie (1981) reports that 3 million computer terminals were installed in 1979; 7.7 million more computer terminals will be installed in 1984; during the 1980s, the workplace will need 30% more computer specialists than there are presently.

For basic reading skills instructors, the proliferation of computers compels us to consider whether there is a relationship between reading comprehension and computer use. We may want to consider this because we are interested in using computers to enhance the teaching of college reading skills, or because we want to help students to participate more effectively in the information world by improving their language facility and computer literacy. Whatever the reasons, we need first to establish the significance of reading comprehension skills to computer literacy and begin to identify those reading skills which are involved in computer use.

Studies of Computer Course Success

What are the aptitudes involved in success in computer courses? Typical early studies such as those by Petersen & Howe (1979) found a highly significant correlation between success in an introductory computer course and high school grade point average, class rank, GPA in math and science courses, college GPA, and general intelligence. Certainly many basic skills students do not rank high in their high school graduating classes, nor do they have high GPAs generally, or in math or science courses. In addition, the Petersen & Howe study only indirectly links success with computers to reading comprehension: good reading skills are usually an important component in high GPA.

In another study, Alspaugh (1971) used a programmer aptitude test consisting of completion of number sequences, comparison of geometric pairs, and completion of word problems, to determine if there was a relationship between grade in school, programming aptitude, and programming achievement. He found that the programmer aptitude test scores of the college students were significantly higher than those of the high school students (college $\bar{x}$ = 56.39; h.s. $\bar{x}$ = 45.07). The investigator then sought to see how altering instruction would affect achievement. Although the content of both courses was the same, college students were given one hour of instruction in introductory FORTRAN each week, high school students two hours along with a greater number of programming examples. Both sets of students received the same 2-hour final exam. Alspaugh found that both groups were equally successful with the course. This

study is particularly interesting for those of us who teach basic skills students. It would suggest that when there are significant differences in aptitudes, achievement can be realized through altering the instruction without altering the content.

Although studies such as Alspaugh's completed in the 1970s showed the Programmer Aptitude Test to be a good predictor of success in computer courses, more recent studies report these same tests as low predictors of success in introductory computer courses. Instead, Wileman, Konvalina, and Stephens (1981) investigated other factors using a five-component predictor test consisting of reading comprehension, alphabetic and numeric sequences, logical reasoning, algorithmic execution, and alphanumeric translation items. The correlations found were more significant than those relating Programmer's Aptitude Test with final exam score. The authors suggest, based on their findings, that students who do not score well on the predictor test register for a less technical "computer appreciation" course; if successful at that level, they might then be allowed to register for regular computer courses. Other conclusions might include improving students' skills with the five components tested, since not only did these variables (reading, sequencing, logic, algorithms, and translation) correlate significantly with the final exam, they correlated with each other.

Instructor and Student Assessments

To further confirm the relationship between the level of reading comprehension and achievement in computer courses, I interviewed the Chairman of the Computer Science Department at Hunter College. According to Professor Thomas Wesselkamper, more students fail introductory computer courses at Hunter College because of problems with communication skills -- reading and writing -- than for any other reason, including math deficiencies. After a student writes a computer program, he must document it, i.e., write an explanation understandable to others who may use or alter the program. He must also be able to read documentation written by others in order to execute or change programs. Reading and writing documentation depends to a considerable degree on precision of meaning, requiring accurate literacy skills. According to Professor Wesselkamper, students also seem to have great difficulty recognizing ambiguities, thinking and solving problems systematically, and sequencing ideas, all critical to creating programs. All of the skills which Professor Wesselkamper described as important to computer use are usually addressed in college basic reading skills courses.

In addition, I interviewed six students presently taking a college reading skills course who are also enrolled in or have recently been enrolled in introductory computer courses. Four of the six students are non-native Americans for whom English is a second language. When asked to describe what they viewed as most important to doing well in the introductory computer science course, all six indicated being able to read and understand the textbook as most important followed by taking accurate notes on both textbook and class lectures. Only one of the six thought a computer science background was important; five felt that an advanced math background was not required; all six referred to the necessity of logical thinking (certainly emphasized in the introductory courses at Hunter). Only one student mentioned using the computer as important to doing well in the course. Other factors mentioned included understanding the instructions for completing assignments.

Computer Assignments and Documentation

Examining computer course assignments reveals further the necessity for accurate reading comprehension skills. The stated problems assume that the student brings varied and prior knowledge and the ability to apply it to the task. However, if the student does not know the meaning of the words or concepts involved in the task, then, according to Burns (1983), he will not be able to conceptualize it in order to follow instructions to solve the problems. Even if the student knows the algorithms, or formulas, for solving problems, but the material surrounding these is dense reading, the student may not be able to imagine what he should do. On the other hand, if he does not know the algorithms, he cannot place the task within a context. It would seem then that the same skills central to the reading process, manipulating grammatical structures, unpacking dense reading material and breaking it down into chunks, understanding vocabulary, and bringing prior knowledge to the text, are also critical to successfully using the computer.

Certainly the skills described so far as important to computer use are also important to reading comprehension. However, according to Guthrie (1981), the cognitive demands of using the computer are different from those of reading textbooks; he contends that these different

demands are governed by certain rules which can be taught. For the task of text editing, the skills necessary include being able to participate in a highly interactive process requiring considerable reading skill as well as determining goals, models, methods, and selection rules. While reading was necessary and at the center of the task, it was not sufficient. Other cognitive demands were tied to the reading process.

In a related study, Miller (1982) examined the types of materials read by technical and managerial staff in a high-technology industry. The materials involved complex concepts, information, and data. According to Miller, the five top reading problem areas perceived by the workers included:

> interpretation, acronym/terminology problems, formatting of material, comprehension difficulties, and conditions at work that affected their reading. (p. 114)

Miller suggests strategies to help with these problems: a "system-thinking approach," reading one another's writing to understand what helps and hinders in reading, using technologically related reading materials for classroom practice, determining the structure of the material, learning specific strategies to locate and interpret key points, and managing reading time efficiently. Most of these techniques are presently part of college reading courses; those which are not could be integrated into existing curricula.

An Optimistic Conclusion

Computers appear to be playing an increasing role in our society. For those who are literate and well-educated, working with computers may initially present some problems based on anxiety and fear of loss of control to or dependency on a machine; these will by and large be overcome through a sense of the computer as a useful tool with which to solve problems, and create, store, and access information more quickly and efficiently.

However, for basic skills students for whom the printed word presents enormous obstacles, attempting to achieve computer literacy is an additional burden. As a basic skills reading instructor, I am increasingly concerned that the task of catching up for my students approaches the myth of Sisyphus; as fast as they run, they always seem to lag behind.

These studies and interviews suggest that there is a relationship between reading comprehension and computer use. Therefore, the skills we teach in college reading skills courses have an added significance for those students who want to become computer literate; perhaps we can help them to achieve greater computer literacy through improving their reading comprehension.

REFERENCES

Alspaugh, J.W. The relationship of grade placement to programming aptitude and FORTRAN programming achievement. Journal for Research in Mathematics Education, 1971, 2, 44-48.

Burns, G. Children's approaches to programming: A teacher's perspective. Paper presented at the meeting of the American Educational Data Systems Association, Portland, Oregon, May, 1983.

Guthrie, J.T. One computer literacy skill. Journal of Reading, 1981, 24, 458-460.

Miller, P.A. Reading demands in a high-technology industry. Journal of Reading, 1982, 2, 109-115.

Petersen, C.G., & Howe, T.G. Predicting academic success in introduction to computers. AEDS Journal, 1979, 12, 182-191.

Wesselkamper, T. Personal communication, April 6, 1983.

Wileman, S., Konvalina, J., & Stephens, L.J. Factors influencing success in beginning computer science courses. Journal of Educational Research, 1981, 74, 223-226.

Needs Assessment, Instructional Design, and the Use of Microcomputers for Instruction

David F. Salisbury

Florida State University

Abstract

This paper outlines a decision making process which might be useful to people deciding when, where, and how to use microcomputers for instruction. This decision making process involves the phases of needs assessment, instructional design, and media selection and shows how these phases relate to the decision of whether or not to use microcomputers for instruction.

Needs Assessment, Instructional Design, and the Use of Microcomputers for Instruction

When a classroom teacher, academic department, or educational institution considers the decision of whether to use microcomputers for instruction they may not have in mind a definite process to use in helping them make that important decision. Often the decision to use microcomputers for instruction is made without a specific purpose or objective in mind. If no one states specifically what the microcomputer is supposed to contribute--what problem is it intended to solve--everyone involved with the program will be expecting something different and the microcomputer is bound to fail to meet everyones diverse expectations. Also, the absence of any statements which make known the intended or expected effect of the microcomputer make it unlikely that any data will be collected to evaluate whatever contribution it does make.

The purpose of this paper is to outline a decision making process which might be useful to people deciding when, where, and how to use microcomputers in a given instructional setting. The decision making process which will be recommended is based on three distinct phases. These phases are: (a) needs assessment; (b) instructional design; and (c) media selection. The basic questions which correspond to these three processes are:

1. What student performance gaps exist?
2. What functions need to be performed in order to close the student performance gaps?
3. What alternatives are available to perform the needed functions given constraints X, Y, and Z?

The Needs Assessment Phase

The answer to the first question lies in the realm of needs assessment. Sources such as Burton and Merrill (1979) and Kaufman and English (1978) outline simple procedures for conducting such assessments. My purpose here is not to describe in detail how to conduct a needs assessment. For this, one should consult the sources mentioned above. What I do want to show is how the output from a needs assessment serves as input to the instructional design and media selection phases.

Essentially, the output of a needs assessment is a list of "needs statements" which identify "gaps" in student performance by indicating how students are presently performing and comparing that with how they should be performing. Some examples of needs statements are:

1. "The average student score on a course final exam is 64. We want to increase the average score to at least 80."

2. "Fifty percent of the students have a negative attitude toward the Social Science 100 survey course. We want to reduce that percentage to no more than 10%."

As you will note, each of these needs statements consist of two parts: (a) a statement indicating the results we are currently obtaining, and (b) a statement indicating the results we want to obtain. The difference between the current results and the desired results is the student performance "gap". Notice that needs statements are not statements such as "We need microcomputers in our school", "We need to reduce the paperwork burden on teachers", or, "We need to have computer-assisted instruction". These statements are in reality solution statements rather than needs statements because they talk about the solution rather than about the problen. The needs assessment phase simply serves to identify what the problems are in terms of student performance gaps. The student performance gaps become the input to the instructional design phase.

The Instructional Design Phase

Once we have identified gaps in student performance we are ready to enter the realm of instructional design. Instructional design addresses the question, "What functions must be performed in order to close the student performance gaps?" One possible model which might assist us in determining the needed functions is Gagne's Events of Instruction because this model has been substantiated by a large amount of research (Gagne & Briggs, 1979; Gagne, 1977). This model suggests nine events which must take place as part of the instruction in order for learning to occur. Table 1 lists these events and provides a brief description of each.

Insert Table 1 About Here

Providing for these events more effectively in the instruction would probably close the student performance gaps. The objective, then, of using this model would be to identify events which are weak or missing in the instruction.

Suppose for example that we want to address the need identified by the first need statement related to increasing the average student score on the course exam. We should now ask the question, "What functions need to be performed in order to bring up the scores?" We might determine, for example, that the instruction is not providing the students enough practice with feedback (a combination of Events 6 and 7). This then, would be our answer to the second question. Providing students with more practice and feedback would probably bring up the scores.

In summary, what we should end up with at the conclusion of the instructional design phase is a set of statements prescribing functions which need to be performed as part of the instruction. These functions are those which the instructional design literature suggests will contribute to closing a particular student performance gap. Notice that we have still not prescribed a particular medium, device, program, or set of materials. We have only identified functions to be performed. There may be various ways in which the functions might be carried out. At this point we are ready to move to the next phase--media selection.

The Media Selection Phase

The media selection phase addresses the question, "What alternatives are available to perform this function given constraints X, Y & Z?" In other words, we now want to determine what means to use to perform the functions identified in the previous phase. Constraints such as time, money, and human resources will need to be considered here.

Some media selection models exist which relate various media to functions to be performed as part of instruction. (Briggs & Wager, 1981; Reiser & Gagne, 1983). For example, suppose the function to be performed is that of providing additional practice with feedback. One possible means of providing practice with feedback is microcomputers. Other alternatives would include flashcards, working with a tutor, or practicing with peers or with parents. Especially in the area of practice, microcomputers have several advantages and capabilities over other alternatives. Merril & Salisbury (1983) and Siegel and Misselt (1983) have shown that microcomputers may well be the ideal medium for providing practice because they can utilize very sophisticated practice paradigms and review items in a systematic way. However, the critical question to ask in determining if microcomputers be used to provide practice is if the software is available for the needed application and if that software employs a strategy

superior to that which could be done by flashcards drills with a classmate or parent. In other words, only if the unique capabilities of the microcomputer can be used to better perform the necessary function can it be expected to produce better results than some other medium.

Microcomputers can also be used in a testing situation to provide additional practice and feedback. Computer generation of multiple test forms and diagnostic feedback which can be furnished by microcomputers allow students to take a test, get immediate diagnostic feedback, study further on specified objectives, and then retake a test. Any of these alternatives would probably contribute to closing the student performance gap if the cause of the problem is lack of practice with feedback.

As a second example, suppose that we determine that the student performance gap is due to the students' inability to retain the information and transfer their learning to new situations. Our answer to the second question, "What function needs to be performed in order to close the student performance gap?" would be to provide a strategy for helping the students remember and transfer the information (Event 9).

We would then ask the third question, "What alternatives are available to perform this function given constraints X, Y and Z?" Here, the alternatives would include such things as providing the students with mneumonic devices to help recall the information, presenting the students with additional examples of concepts or rules in other contexts. This could be done by means of board game simulations, field trips, or workbooks. Microcomputers, of course, would be another alternative. They could be used to provide experience in using the information in the form of computer simulations. Another alternative would be to utilize microcomputers to present additional questions or problems requiring application of the information in different contexts. Again, the unique information handling capabilities of the microcomputer could be used to keep track of items presented to individual students and present new items in a systematic way. Only if instructional software is available which would enable the microcomputer to provide this function better than the other alternatives, and if the existing constraints allow, would they be the best solution to the problem.

Microcomputer delivered tutorial/dialog instruction could provide a solution if the student performance gaps can be attributed to the students receiving poor presentations of the instructional material. Some material is difficult for teachers to present in a logical, systematic, step-by-step manner to students who differ greatly in the amount of previous knowledge and preparation. If appropriate tutorial/dialog software can be found or produced to present this problematic material to students on an individualized basis then microcomputers may prove to be the most powerful solution.

Figure 1 summarizes the questions which need to be answered as part of this three-phase decision making process to determine whether microcomputers would provide the best solution to a particular instructional problem. Possible sources for obtaining the information needed to answer each question are given in the middle row of the figure and a description of the output at each phase is given in the bottom row.

Insert Figure 1 about here

The figure gives a representation of the relationship between needs assessment, instructional design, and the selection of appropriate media. Needs assessment alone does not tell one that microcomputers are or are not needed. What needs assessment does do is illuminate gaps in student performance and provide input to the instructional design phase. Instructional design does not determine whether microcomputers should be used for instruction. What instructional design does is identify functions which need to be performed in order to close or eliminate gaps in student performance. Only if microcomputers are selected as being the most appropriate medium for performing these functions should they be expected to produce better results than something else.

Conclusion

The numerous articles appearing in the press about microcomputers are encouraging in the minds of many people the assumption that these new machines will be able to solve many of our educational ills. The potential is certainly there. But as David Miller (1983-84) said in his introduction to a

recent issue of Educational Technology Systems, "there are . . . educators who have seen the disillusionment that has followed previous cycles of optimism with new devices, and they are asking . . . for a careful analysis of just how this new technology can meaningfully impact education" (p. 1). No one should be pressured into using microcomputers for instruction by vendors who are seeking monetary gain by making excessive claims about their products. As Michael Potts (1983) has recently pointed out:

"We educators have gotten off to a poor start, bringing computers to school; our uncertain focus has been distracted by the loud noises of vendors seeking easy megabucks in computer hardware and software. I'm tired of hearing that "Star Wars" teaches geometry and "Lemonade Stand" teaches economics; baseball teaches both much better, and for the price of a bat and ball you can engage a whole class" (p. 22).

As the above quotation suggests, there is no need to purchase microcomputers for instruction if some other medium will work as well or better, and for much less cost. If the decision to use microcomputers for instruction is based solely on the basis of "fadism" or modernization without careful analysis of needs, thoughtful consideration to instructional design principles, and proper media selection the potential impact on education will probably be small. However, if we follow this three-phase decision making procedure and base our decisions to use microcomputers for instruction on needs in terms of student performance gaps we stand a better chance of making a difference in the quality of a course or program that can be seen and agreed upon by all.

References

Briggs, L. J., and Wager, W. W. The handbook of procedures for the design of instruction. (2ed ed.). Englewood Cliffs, N.J.: Educational Technology, 1981.

Burton, J. K., and Merrill, P. F. Needs assessment: goals, needs, and priorities. In L. J. Briggs (Ed.), Instructional design: principles and applications. Englewood Cliffs, N. J.: Educational Technology, 1977.

Gagne, R. M. The conditions of learning, (3rd ed.). New York: Holt, Rinehart and Winston, 1977.

Gagne, R. M. & Briggs, L. J. Principles of instructional design. (2ed ed.). New York: Holt, Rinehart and Winston, 1979.

Kaufman, R., and English, F. Needs assessment: concept and application. Englewood Cliffs, N.J.: Educational Technology, 1979.

Merrill, P. F., and Salisbury, D. F. Research on drill and practice strategies. Manuscript submitted for publication, 1983.

Miller, D. C. Editorial overview. Educational Technology Systems, 1983-84, 12, (1), 1-3.

Potts, M. Still? Educational Computer, 1983, 3, (5), 22-23.

Reiser, R. A., and Gagne, R. M. Selecting media for instruction. Englewood Cliffs, NJ: Educational Technology, 1983.

Siegel, M. A., and Misselt, A. L. An adaptive feedback and review paradigm for computer-based drills. Journal of educational psychology, 1983, 76(2).

Table 1

Nine Events of Instruction

1. Gain attention.
2. Inform the learner of the objective.
3. Stimulate recall of prerequisite learning.
4. Present the stimulus material.
5. Provide learning guidance.
6. Elicit the performance.
7. Provide feedback about performance correctness.
8. Assess the performance.
9. Enhance retention and transfer.

Needs Assessment	Instructional Design	Media Selection
Question: What student performance gaps exist?	**Question:** What functions need to be performed in order to close the performance gaps?	**Question:** What alternatives are available to perform the needed functions given existing constraints?
Source of information: Needs Assessment	**Source of information:** Instructional Design Literature	**Source of information:** Media Selection Models
Output: Needs Statements	**Output:** List of functions to be performed	**Output:** Best alternative means for closing the gaps

EDUCATIONAL SOFTWARE FOR PROBLEM SOLVING

BY THE HEARING IMPAIRED STUDENTS

Manjula Waldron, Assistant Professor
Department of Engineering Graphics
The Ohio State University

Susan Rose, Assistant Professor
Department of Educational Psychology
University of Minnesota

Steven Heinrich, Student, OSU

Abstract:

Problem solving is an important skill which must be learned to be a successful adult in today's world. Hearing students usually acquire these skills in the normal course of instruction. However, greater reliance on rote learning and communication difficulties in hearing impaired classrooms result in a low level of problem solving skill in most hearing impaired students. Congenitally strategies employed during this intellectual function by the normal population is based primarily on spoken language and hearing. For the cognitially profoundly hearing impaired, whose hearing, and hence spoken language is impaired, different perceptual strategies may be in use during problem solving. In this paper, software important to problem solving is examined and its adaptations for the hearing impaired is presented. Any software developed should be easy to use by both the instructor and the student.

Specifically a computer program called CUBES was developed with this in mind. The object of this program is to teach visual spatial estimation. This software has three parts: a program through which teacher can create the visual spatial task to estimate the number of basic blocks which have been used to create the shape shown. Once created the teacher can either put these in a train or a test file. The entire program is menu driven and is extremely easy to use. The technique capitalizes on the use of visual-graphics symbols with the use of formal English language minimized to suit the specific learning style of this population.

Introduction

George Polya (1957) in his book How to Solve it very concisely states the importance of providing any student with an opportunity to solve problems: "A great discovery solves a great problem but there is a grain of discovery in the solution of any problem. Your problem may be modest, but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your own means, you may experience the tension and enjoy the triumph of discovery. Such experiences at a susceptible age may create a note for mental work and leave their imprint on mind and character for a life time." Thus, the teacher of deaf students can either challenge the curiosity of their student or fill the time with drill and practice. To challenge however, the teacher must set tasks appropriate to the student's knowledge and assist him/her in solving problems with appropriate means to give him/her a taste for independent thinking which is crucial to problem sovling.

Normal hearing students learn problem solving skills in the normal course of their education. This occurs since the students are given a greater degree of autonomy in the classroom, school day, and at home. Easy communication and incidental learning seem to assist these students gaining the confidence of their teachers, peers, and community. It is a kind of snowballing effect; the more opportunities they get to solve problems and make independent decisions the better they become at it. On the other hand, both the classrooms for the deaf and the traditional teaching philosophy of the deaf provide the students with little autonomy. This may be due to communica-

tion difficulty, low vocabulary levels and a total emphasis on teaching English language and speech to the deaf. The only methods which seem to succeed with the deaf in teaching English language and speech are rigid and imitative in their delivery; like the Apple Tree curriculum, the Fitzgerald key and Ling method to name a few. If the students digress from the pre-laid paths which they must follow, they encounter negative reinforcements at every step. This imitative process of learning provides little opportunity for the deaf student to solve problems or make decisions. It is not surprising that the majority of deaf students become passive learners and lack the essentials of problem solving.

The use of visual imagery as a form of symbolic representation is a critical aid in variety of cognitive skills. It is a representational aid in memory, abstraction, prediction and hypothetical-deductive thinking (Kosslyn, 1980). The ability to mentally manipulate images is essential for such spatially-oriented subject matter as mathematics, geography, art, and other natural science (Papert, 1980). This is especially true in problem solving situations involving prediction and anticipation of events, particularly when they require transformation. Ottem (1980), in his analysis of cognitive studies in comparing hearing and deaf populations, found similarities between the two groups at the concrete operational stage of development. Imagery has been used as an instructional tool with deaf children (MacDougall, 1979), but these techniques have been applied primarily to word acquisition tasks such as paired-associate learning. Klein (1983), in a pilot study, presented kinetic imagery tasks to deaf and hearing adolescents and found the deaf to be significantly superior in arc and angle transformation tasks. Visual imagery may thus play an important role in problem-solving strategies employed by deaf persons. However, not much work has been reported on how deaf children solve problems. This may be due to the fact that, until recently, it was assumed that good English reading and language skills were precursors to problem solving. The advent of computers and computer graphics allowed problems to be presented through means other than spoken or written language. Rose, Waldron & Kolomyjec (1983a) asserted that it was possible to develop problem solving skills when the barriers imposed by English language were reduced.

Waldron and Rose (1983) designed a series of non-verbal thinking tasks including spatial relations, pattern recognition, matrix board and inferencing using computer graphics. The tasks were designed so that no use of English language was necessary in solving the problems presented to the students. Once the use of equipment was demonstrated the student could proceed using his or her own problem solving style. In a pilot study they reported little success bias between hearing and deaf individuals but there was a definite difference in the approach to the solution of the problem. The hearing students sought English clues and explanation while the deaf proceeded without these. This software has since been adapted for microcomputer use.

Microcomputers and Problem Solving:

For the first time, microcomputers with graphics have placed a teaching tool in the classroom that can be used effectively to allow the student to work within his or her own rule system and to attack a problem with the language barriers removed. The development of user oriented software where directions may be graphically presented and simplified and consistent allows the hearing impaired learner to be creative, to deduce and to internalize the consistency of computing languages and the teachers can produce these readily in the classroom (Waldron & Rose, 1981; Rose & Waldron, 1983a).

Over the past six years the influx of computers, in the classroom for hearing handicapped children has grown at astounding rates. A survey conducted by Harding & Tidball (1982) regarding the use of computers and microcomputers in schools for the deaf documented the beginning of the movement. In 1980-81 school year microcomputers in use were in two schools and the rest were distributed in thirty-four school systems. The same population was surveyed during the 1982-83 school year by Rose and Waldron (1983b). Microcomputers were evident in classrooms for the hearing impaired in 50 states. One hundred and twelve programs (52% of the programs surveyed) had access to microcomputers for the instruction of hearing impaired students. Forty-five programs stated that microcomputers were requested for the 1983-84 school year. Consistent with the earlier data from Harding & Tidball's research, the primary use of microcomputers was in the area of drill and practice in mathematics and in the area of language and reading. The Rose & Waldron survey further indicated

that teachers of the hearing impaired were generally in favor of the introduction of computer technology into their classroom but were hesitant due to a lack of training in computer use and the lack of software that was directly applicable to hearing handicapped students.

Teaching of general problems requires one to examine the interaction of the real world and the abstract symbolic system used. For the deaf this symbolic system becomes largely visual. Studies in the field of artificial intelligence have had an effect on the educators view of the problem solving behavior. Researchers have developed computer based activities with the stated goal of improving student's performance on problem solving (Kraus, 1982). Learning by doing takes on a new meaning in the interactive computer graphics environment especially for the hearing impaired where a rapid feedback allows the student to reshape and reconstruct his own responses. Well designed computer software can help integrate learning of concepts and logical process inherent in them. Microcomputers have been found to be an effective interactive medium for hearing impaired persons (Goldenberg, 1979).

Computer Software for the Hearing Impaired Students

There is very little computer software that is specifically designed for the hearing impaired. Existing software is predominantly restricted to drill and practice in the area of language and mathematics. Most of the limitations are due to availability of lack of interactive software until recently with graphics capabilities for this population. However, some software utilizing many aspects of the graphics capabilities of computers is beginning to emerge for normal students, which may be relevant. Linehan's Taxi game involves students in problems concerning routing through a three-dimensional city. The Learning Company's logic programs developed by Piestrup share with many of the popular adventure games the movement through a matrix of rooms in the completion of tasks. Programs developed by Bolt, Beranek and Newman use tree diagrams to illustrate or symbolize the structure of stories. The TABS-Math Project at The Ohio State Univesity uses dynamic movement of objects in the illustration of both geometric and probablistic phenomena. Numerous examples of courseware (notably Sharon Dugdale's Green Globs) involve students in activities related to graphing in two dimensions. It should also be noted that many courseware sequences use graphics as an attention-getting device or even to "reward" correct answers.

The software which has been specifically designed and used or adapted for use by the hearing impaired falls under two main categories: Programs for use by students and in the form of authoring languages for use by teachers and students.

1. Programs for student use:

a. Drill and practice: Much of the early CAI material fell into this category and was quickly adapted for use on the microcomputer for normal students. Programs like MATCH MACHINE, SPELL, SUPERMATH, HURKLE ETC. have been developed for microcomputers. These programs have been used with some hearing impaired with isolated adaptations for the student by teachers (Stacy, 1982). The Educational media production project for the hearing impaired (EMPPHI) at the University of Nebraska made language adaptations for NUMBER, HURKLE and BAGELS geared for intermediate, junior high hearing impaired students (Propp, Nugent and Tidball, 1982).

b. Games: Some games have been utilized for pure enjoyment. Other games focus on fast reactions, involving hand eye coordination through keystrokes, paddles or joystick controls. Some games invoke problem solving and decision making skills. Commercial software examples including HANGMAN, STAY AFLOAT, SINK THE SHIP, SPACE INVADERS, STAR BLAZERS, STAR TREK, PAC MAN, MATHQUIZ, AND ESTIMATION, MATHQUIZ FOR 2 are available in various categories for normal students. EMPPHI developed SPACESHIP LOST for intermediate students. This program is based on a Piagetian framework in spatial and directional concepts. MECC software called OREGON had its language level adapted to suit the population.

2. Authoring Languages

a. Some Examples of Existing Software: It is becoming easier for teacher and students to become involved in producing or modifying computer instructional materials. Some of the

languages that are specifically useful to teachers are SUPER PILOT (based on UCSD, PASCAL), author for problem solving and visual thinking, (Waldron Rose & Barber, 1983), BLOCKS (Slovik, 1982), LOGO (Pappert, 1980), and Drawing (Kolomyjec, Waldron & Rose, 1983). The teachers can program predesigned lessons in SUPER PILOT, or they can let the students engage in exploratory learning through LOGO. In the problem solving and visual thinking software the teacher can design the tasks for students use. The Drawing program students can create their own artwork using a joystick. This was successfully used by the hearing impaired students (Rose, Waldron and Kolomyjec, 1983b).

b. Cubes Software: A new software called CUBES to teach geometrical and spatial estimation concepts was designed specifically for use by the teachers and the hearing impaired students. The chief attraction of this software is that it is extremely user friendly. The code is written in BASIC for Apple II+ microcomputer and consists of three parts: 1) The create program allows the teacher to create the problems for their students. They can create a geometric shape by removing any of the 27 minicubes which form a large cube (Figure 1). The student's task is to estimate the number of the remaining cubes. The teacher can create a training file (Figure 2) for use in teaching the student these concepts or a testing file to test if the students have realized a prestated level of achievement. The entire software is menu driven where the teacher can very easily select from the displayed menu whether she wants to create a problems file, initialize a diskette, run a demo program or exit the program (Figures 3a, b and c).

Summary:

There is increasing evidence that the deaf can acquire problem solving skills quite adequately provided they are given the problems when the language is within their grasp and the presented problems use largely graphics material. In this paper general problem solving techniques as used by the hearing impaired students are discussed. How these can be utilized by the hearing impaired students on a microcomputer is examined. The software which is currently available for use by the hearing impaired is reviewed. A new teacher authoring software called CUBES aimed at teaching geometrical spatial concepts is presented. This software is user friendly and is designed for uses by the teacher to present new problems to their students either to train them in this concept or to test their achievement.

Acknowledgement:

This project was partially funded by a grant from the Department of Education personnel preparation program for the handicapped #029CH20103.

References:

Goldenberg, E. P. Special Technology for Special Children. University Park Press, Baltimore, MD, 1979.

Harding, R. H. & Tidball L. K. A national microcomputer software survey for microcomputer usage in schools for the hearing impaired. American Annals of The Deaf. 1983, 127, 5, 673-683.

Klein, E. L., The development of mental imagery in deaf and hearing adolescents, Presented at AREA Meeting, Montreal, 1983.

Kolomyjec, W., Rose, S. & Waldron M. B. Computer graphics and creativity/problem solving skills with deaf and severely language disordered students, Part III: Using microcomputer as a creative tool. Proceedings CEC National Conference on Special Education. Nazzro (Ed) ERIC/CEC, Reston, VA 1983.

Kosslyn, S. Image and Mind. Cambridge MA: Harvard University Press, 1980.

Kraus, W. H. Math learning games. School Science and Mathematics. 1982, LXXXII (5), 397-398.

MacDougall, J. C. The development of visual processing and short term memory in deaf and hearing children. American Annals of the Deaf, 1979, 124, 16-22.

Ottem, E. An analysis of cognitive studies with deaf subjects. American Anals of the Deaf. 1980, 125, 564-575.

Papert, S. Mindstorms. New York: Basic Books, 1980.

Polya, G. How to Solve It. New York: Doubleday, 1957.

Propp, G., Nugent, G. C., Tidball, L. K. Educational media production project for the project for the hearing impaired: Project update. American Annals of the Deaf, 1982, 127, 5, 685-689.

Rose, S., Waldron, M. B., & Kolomyjec W. J. Computer graphics and creativity problem solving skills with deaf and severely language disordered student: Part I. To appear in microcomputers in special education ERIC/CEC, Reston, VA, 1983a.

Rose, S., Waldron, M. B. & Kolomyjec, W. Measuring creativity through computer graphics on hearing impaired children. Perceptual and Motor Skills, 1983b, 57, 943-950.

Rose, S. & Waldron, M. B. Use of microcomputer in teacher training programs American Annals of the Deaf, 1983a, 128, 5, 553-556.

Rose, S. & Waldron M. B. Survey of microcomputer use attitudes and opinions. Final report: Spencer Foundation, Columbus, Ohio, The Ohio State University, 1983b.

Slovik, O., CAI at CSDF: Microcomputer based authoring system. American Annals of the Deaf, 1982, 127 (5), 536-549.

Stacy, E. M. Students: Handle with CARE. American Annals of the Deaf, 1982, 127,5, 617-624.

Waldron, M. B. & Rose, S. Final report: Model development on the inductive reasoning and decision making process of prelingually hearing impaired persons through use of computer graphics. Spencer Foundation, The Ohio State University, College of Education, 1981.

Waldron, M. B., Rose, S. & Barber, F. Computer graphics and creativity/ problem solving skills with deaf and severely language disordered students. Part II. Proceedings CEC National Conference on Use of Microcomputers in Special Education. Nazzaro J. (ed.) ERIC/CEC, Reston, Va, 1983.

Waldron, M. B. & Rose, S. Visual thinking for severely language handicapped children through the use of computer graphics. Journal of Computer Based Instruction, May 1983, 9, 206-210.

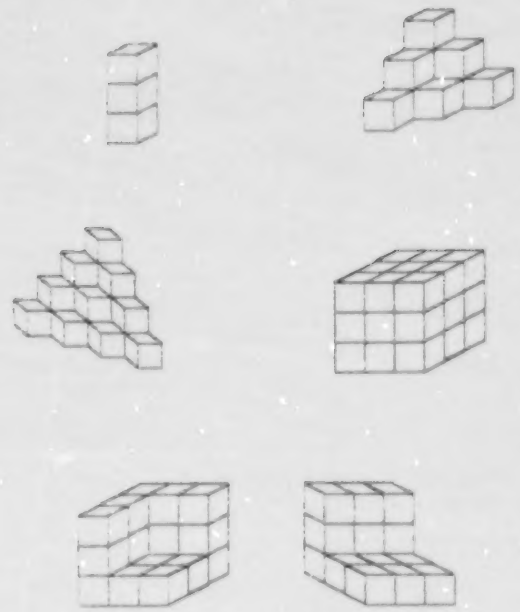
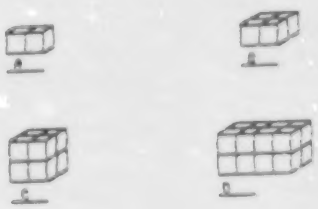


Figure 1. Samples of shapes teacher can create.



HOW MANY CUBES IN FIGURE A ? 2
GOOD WORK !

HOW MANY CUBES IN FIGURE B ?3

TRY AGAIN

HOW MANY CUBES IN FIGURE B ?5

TRY AGAIN

HOW MANY CUBES IN FIGURE B ?4

YOU HAVE NOT BEEN SUCCESSFUL. THE ANSWER IS 4

HOW MANY CUBES IN FIGURE B ?4

CORRECT !

Figure 2. Sample teacher designed training task and response.

BEST COPY AVAILABLE

```
            MENU
            ----
 (1) CREATE TEST/TRAIN FILES
 (2) INITIALIZE NEW DATA DISKETTE
 (3) RUN TEST PROCEDURE
 (4) RUN DEMO
 (5) PRINT STUDENT FILE
 (6) CATALOG DATA DISK
 (7) EXIT
 CHOOSE ONE
```

Figure 3a. Teacher program menu.

```
 PRESS ANY KEY TO CONTINUE
          O   I   O
          \   |   /
           \  |  /
            \ | /
          J---+---L
             /|\
            / | \
           /  |  \
          M   .   .
                    POINT MOVEMENT
                      DIAGRAM

P - PLOT LINE BETWEEN TWO POINTS
E - ERASE LAST LINE DRAWN
T - TOGGLE BETWEEN THIS PAGE AND HGR
Q - QUIT DRAWING SHAPE
M - MOVE TO NEXT POINT WITHOUT DRAWING
F - 1/4 LENGTH OPTION
R - RESTORE DIMENSIONS TO FULL LENGTH
```

Figure 3b. Teacher shape create menu.

```
                          CUBE     DEFAULT
   --------               SIDE     VALUE
  /       /|
 /       / |
 --------  |              A        25
 |       |  |              I        25
 |       |  /  ANGLE      J        20
 |       | /
 |-------|/               ANGLE    30 DEG.

WISH TO CHANGE ANY VALUES (Y,N)
```

Figure 3c. Cube parameter set-up menu.

BEST COPY AVAILABLE

THE EVOLUTION, EXECUTION, AND EVALUATION OF A "COMPUTER LITERACY" WORKSHOP FOR HEALTH EDUCATORS

Moon S. Chen, Jr. The Ohio State University

Sheila R. Brill The Ohio State University

ABSTRACT

To meet the need for "computer literacy" among health educators, the authors-instructors developed, taught, and evaluated a three credit summer workshop for graduate and undergraduate students at The Ohio State University. The overall goal of this workshop was to provide students with a cognitive and a practical base for asking and answering questions relating to the state-of-the-art in microcomputer-based health education. This overall goal was accomplished through a variety of educational techniques including extensive hands-on time with microcomputers, lecture-discussions, readings, and handouts, demonstrations of the capabilities of the mainframe and mini computers, tours of computer facilities, guest speakers, and opportunities to evaluate currently available computer-based health education software. As a result of this workshop participants were better equipped to address both the philosophical and pragmatic aspects of such issues as the utilization, role, effectiveness, and future of microcomputer-based health education. Very favorable and valuable student feedback of the workshop were received. Based on student feedback and the authors-instructors' own assessment, a model for future introductory level "computer literacy" courses for health educators was tested. Next steps for improving this model were recommended.

Despite the ambiguity of its meaning, "computer literacy" has been increasingly identified as a need for health educators. This need has been manifested by informal conversations among members of the Health Education Special Interest Group at the 1983 ADCIS Conference in Denver and special editions of journals devoted to computer utilization for health professionals (e.g. Journal of Computer-Based Instruction, Vol. 9, No. 3) and for health educators (e.g. "The Microcomputer and Health Education", October 1983 issue of Health Education). Another means of meeting this need has been a greater demand for "computer literacy" courses or workshops.

In fields other than health education such "computer literacy" courses or workshops have usually included sessions on computer technology and sessions on applications to the respective field being addressed. An example of such a course in the humanities and social sciences was described by Tannebaum (1983). Other "computer literacy" projects have been conducted for college freshmen (Gerhold, 1983; Friedman, 1983) and for faculty development (Swadesh and Wagstaff, 1983). Few, if any such "computer literacy" courses or workshops have been described in the professional literature for health educators.

Yet the need for "computer literacy" (however defined) is no less for the health educator. Possibilities for utilizing the computer in the health professions, specifically in health education have barely been realized. Perhaps the major barrier in such utilization has been the dearth of health educators prepared to take full advantage of computer technology for health education.

Consequently, during August 1-5, 1983, a three credit hour workshop (for both graduate and undergraduate credit) was piloted at The Ohio State University. This workshop was entitled "Health Education Possibilities with Microcomputers" and was a "hands-on" laboratory and discussion workshop to examine and evaluate

currently available health education software for microcomputers. No previous knowledge of microcomputers was assumed.

In this paper the authors will further describe the evolution of this workshop, outline the "execution", i.e. the actual implementation of the workshop, and delineate the students' feedback of this course. Finally, the students' feedback and authors-instructors' analyses will be composited for an overall evaluation of the workshop.

EVOLUTION OF WORKSHOP

The evolution of this workshop began with an informal needs assessment. The four factors addressed in this needs assessment were the objectives for the workshop (what and why), target audience for this workshop (who), availability of facilities (where), and time limitations (when).

What are the objectives of this workshop? (what and why). These objectives were generated based upon our conception of what health educators functioning in either a clinical, community, workplace, or classroom setting would need to begin an acquaintance with computers. Based on this conception, thirty-four objectives were identified. A summary of these objectives are listed in Table 1.

Who is the target audience? (who). We assumed workshop participants would have no previous experience with computers. We advertised the workshop as such and subsequently attracted students who were health professionals mostly with a health education background. The workshop could be taken for graduate, undergraduate, or continuing education credit. The number of participants should be limited according to the availability of microcomputers. No more than two participants should be at one terminal.

What facilities are available? (where).

We were very fortunate to have access to the College of Education's Microcomputer Laboratory. This laboratory has nine Apple II Plus microcomputers. Besides the opportunities for students to have "hands-on" experience with the computers, these facilities allowed us to demonstrate a program on one computer and have that program displayed on all the monitors.

With the increasing portability of microcomputers, guest speakers were able to bring their computer demonstrations to the classroom. The students were thus exposed to a variety of microcomputers such as the Compaq and Osborne in addition to the Apple II Plus.

We also had the opportunity for the students to tour selected University computer facilities. These included the main Instructional Research Computer Center (main frame computer), the Division for Computing Sciences for Medical Education and Research computer terminals (for computer-assisted instruction in the health sciences), and the College of Education Microcomputer Laboratory (minicomputer).

What are the time limitations? (when).

The availability of the Microcomputer Laboratory restricted our choice of dates and hours for the workshop. With the ever-increasing interest in microcomputer technology, the Microcomputer Laboratory was booked months in advance. The week of August 1-5, 1983, was the only available time during the summer quarter. That meant that the workshop participants would meet for five and one-half hours each day from 12:00N-5:30 PM. These dates and times also coincided with the annual Health Educators' Institute in Ohio. Unfortunately as a result of this conflict many prospective participants could not attend.

EXECUTION OF WORKSHOP

Both instructors for the workshop (also the authors) were a university health education faculty member and a health education graduate teaching associate. They were "turned on" to microcomputers and had learned from a non-programming perspective the applicability of computers for health education. Consequently, a broad and pragmatic approach to computer utilization was emphasized. In addition guest speakers were invited to share their expertise in various applications of computer-based health education programs. These guest speakers included a nationally known CAI expert (Jeanne L. Burson), a representative from Compuserve, a health software salesman, and other individuals.

The objectives for this three credit hour workshop (see Table 1) were accomplished through a variety of educational techniques including extensive "hands-on" experience with microcomputers, lecture-discussions, readings and handouts, demonstration of capabilities of various microcomputers as well as main frame and mini computers, guest speakers, and opportunities to evaluate currently available computer-based health education software. In keeping with the theme of computers, most of the handouts including the workshop schedule, syllabus, and various lecture notes were generated by computer (CYSPUB/SCRIPT).

The grading of students was based on a vocabulary quiz of computer terminology (20%), two evaluations of available health education software (30%), and a final project (50%). The final project consisted of writing papers in response to two of three options. Briefly, the options were as follows:

1. Develop your own criteria for evaluating a computer-based health education program and defend your rationale for your choice of criteria.

2. Describe your plan for integrating and implementing microcomputer-based health education programs in the professional setting of your choice.

3. Select two issues and address them in depth as they relate to your professional interests. The issues dealt with professional acceptability, desired client level of "computer literacy", cost, and evidence that microcomputer-based programs are efficacious for their delegated tasks.

The goal of these papers was to allow workshop participants both to internalize what they learned and also to relate them more specifically to their professional interests.

EVALUATION OF WORKSHOP

Based on anonymous student evaluations of this workshop, we believed we not only achieved the workshop objectives but exceeded them.

Very favorable written student feedback from this workshop was received on formal course evaluation forms. In quantitative terms students were asked to "grade" their overall impression of this course. On a scale of 0-100, the means for both "grades" was 96 (n=6 and n=7 respectively). These means were the highest the authors-instructors had ever received for any course evaluation.

In qualitative terms students were asked to write their perceptions of the weaknesses, strengths, and any other comments about the workshop. The single major weakness cited was the time constraint and pace of the workshop. The almost six hour per day, five days per week concentrated workshop of thirty hours in duration was apparently too physically draining. On the other hand, the strengths and values of this workshop were many. These were enumerated as:

(a) the "hands-on" or practical experience with microcomputers. (We used the College of Education Microcomputer Laboratory where the ratio of microcomputers to students was almost 1:1.)

(b) very helpful and enthusiastic instructors

(c) excellent handouts given to students and good text

(d) interesting and broad variety of topics (including exposures to computers other than microcomputers), tours to various other campus computer facilities, and speakers.

Among the other comments provided were such statements as "...Thanks I have been turned on to computers." and "...A more extensive programming exercise for me would have been readily accepted". As instructors for this workshop, the authors were extremely gratified by these responses.

Students who took this summer workshop and others have frequently inquired about when this workshop might be offered again. Currently, this workshop is scheduled again for July 25-31, 1984. The next workshop would retain the features identified by students as strengths and incorporate improvements in the scheduling of the workshop.

Our experience with this workshop confirms our beliefs that "computer literacy" for health educators indeed is a set of competencies that is needed and can be taught. What we taught and how it was taught were perceived to be what was needed at the introductory level. Next steps suggested by this experience include replicating this workshop more frequently and following up previous workshop participants to further assist their progress towards "computer fluency".

TABLE 1. WORKSHOP OBJECTIVES*

1. To discuss the historical development of the computer from its beginnings up through the effects of the mass production of microcomputers.
2. To acquaint students with the vocabulary and mechanics (care and use) of computer hardware and software.
3. To provide students with opportunities to review at least four examples of microcomputer-based health education programs and to write up evaluations on two of these programs from their own perspectives of probable utilization requirements.
4. To observe demonstrations by guest speakers of various applications of microcomputer programs in the health care system.
5. To cite evidences from research for the acceptability of microcomputer-based health education programs.
6. To discuss developing a plan to implement microcomputer-based health education programs.
7. To discuss various considerations in purchasing a microcomputer.
8. To define and describe basic concepts in computer-assisted instruction.
9. To discuss the effectiveness of microcomputer-based health education programs.
10. To discuss the apparent theoretical bases for the appeal of computer games and how this appeal might be applied for microcomputer-based health education programs.
11. To discuss the implications of microcomputer technology on the professional practice and preparation of health educators.
12. To study and share insights related to specific topics of their choice (within the boundaries of the Final Project specifications) not otherwise covered in "Health Education Possibilities with Microcomputers".

*Thirty-four objectives for the workshop were actually identified. In this table these thirty-four objectives were summarized into twelve objectives. A complete listing is available on request from the authors.

REFERENCES

1. Tannebaum R S: Introduction to computers for the humanities and social sciences a micro-based computer literacy course. J Computer-based Instruction 9 (Special Issue): 175-182, 1983.

2. Gerhold G: A micro-based literacy course for college freshmen. J Computer-based Instruction 9 (Special Issue): 191-193, 1983.

3. Friedman E: Personal computer plan at Stevens Institute of Technology. J Computer-based Instruction 9 (Special Issue): 270-271, 1983.

4. Swadesh N, Wagstaff N: Faculty computer literacy: A home-based model. J Computer-based Instruction 9 (Special Issue): 266, 1983.

THE ALTERLAB LESSON: A SIMULATION PACKAGE TO TEACH THE PRINCIPLES OF PATTERN ALTERATION

FRANCES WHITAKER MAYHEW

University of Delaware

Abstract

A new instructional package has been developed at the University of Delaware through a grant from the Office of Computer-Based Instruction. The package is entitled Sewing Pattern Alteration Laboratory, "Alterlab" for short. Alterlab is designed to complement classroom instruction by combining a tutorial review of clothing pattern alteration principles with the opportunity to apply these conceptual learnings to practice problems. The package includes a simulation, which means that the students using this program are able to duplicate the procedures they would normally perform to carry out a pattern alteration. However, instead of working with paper, scissors, edge guides and tape; these procedures are performed on the computer screen. In addition, Alterlab provides a judgmental component which evaluates the students' progress through an alteration problem and alerts them to errors or reinforces accuracy at critical points.

For many students enrolling in a college level clothing construction course, the concepts related to pattern alterations are the major new learning. Pattern alterations are adjustments which are made to the paper patterns for garments so that the completed garment will conform to the body proportions and/or contours of a specific individual. Pattern alterations are used when a commercially available paper pattern fails to provide a comfortable or attractively-fitted garment. (It must be remembered that the human body is perfect, it is the pattern that fails to conform!)

One of the most time-consuming tasks the instructor of a clothing construction course must face is the grading of pattern alteration practice problems. Students are usually assigned two sets of alteration practice problems-one for upper body fitting problems and one for lower body fitting problems. There are usually ten problems in each set. At the University of Delaware the enrollment for clothing construction courses is generally eighty students in a Spring or Fall semester. Simple mathematics reveal that the clothing constructions instructor is faced with 1600 pattern alteration problems to grade. Since the students often struggle with these new concepts, they may show extraordinary creativity at layering one erroneous procedure on top of another. Once the students have strayed from the correct alteration procedure, their chances of producing an acceptable solution to the practice problem are almost nil. Untangling the students' (often amazing) incorrect thought processes, demands at least one week of the instructor's time. By the time the assignments are returned to the students, they have forgotten, (and probably no longer care), why they approached a problem incorrectly. This investment of considerable time and effort on the part of both students and instructor often does not produce effective learning. The improvement of this situation was the motivation behind the design of the Alterlab package.

The Alterlab package includes a simulation section which allows the students to perform the operations they see demonstrated by the instructor in the classroom. These operations are used to make the pattern adjustments required to solve the practice problems. In addition, the Alterlab program judges the students'

progress at critical points in the procedure and provides clues to help them correct their errors before allowing the students to continue. These judgments occur at three points in the alteration procedure. The placement of slashlines is judged. The amount of change--be it spread or overlap--is judged. And, the correction of the distortion in the original pattern shape imposed by the alteration procedure is judged. If an error has been made at any of these points, the students cannot correctly complete the problem. Therefore, they are directed to go back and redo the step.

The primary programmer for the Alterlab lesson was Vickie Gardner. The package was developed for the PLATO System. PLATO, a product of the Control Data Corporation, is a computer system designed to be used solely for computer-assisted instruction. The terminal has a touch-sensitive screen which allows PLATO to accept input from the screen as well as from the keyset. The keyset itself is different from a typical typewriter keyboard because it has additional keys on the right side called function keys. These function keys, labeled NEXT, BACK, LAB, DATA, HELP and STOP allow the programmer and designer to create sequencing through the lesson contingent upon the specific key pressed. NEXT and BACK are generally used for moving forward and backward in the lesson, and HELP usually provides remedial support.

The Alterlab package is designed to enhance classroom instruction. It is organized into three main sections. The Tutorial introduction is used to review pattern alteration concepts and can provide remedial support for errors which may be made in the simulation which follows. The Instructions section provides training and practice on the routines which operate the simulation. The Simulation section allows the students to work through a set of assigned pattern alteration problems or perform an individualized alteration. The package also provides a glossary of terms.

TUTORIAL

The tutorial review provides theory and application on three major pattern alteration concepts. It defines categories in which students can classify types of figure variations from the hypothetical "standard". It identifies clues which help students recognize the fitting probkems which are indicated by the presence of these figure variations and it also provides the general alteration procedures which are used to accommodate these figure variations. For example, possible variations from the standard are grouped into three major classifications. The first is variations in LENGTH which include Body Segment Length, such as arm or back waist length; Body Contour Level Location, such as bust level or hip level; and variations in length due to Posture, such as the swayback or debutante slouch. General CIRCUMFERENCE variations which might appear at the waistline, thigh or upperarm girthline comprise the second classification. The third classification groups variations in BODY CONTOUR SIZE which include body contours which are disproportionate in size, such as a full bustline or flat derriere; and body contours which are asymmetrical in size--when one side of the body does not closely duplicate the other. For each of these classifications of figure variations, the fitting problems which are manifested by these variations are discussed. These fitting problems can be recognized as horizontal or vertical wrinkles, pulls or sags, or the misalignment of the grainline. Each of these recognizable fitting problems can be remedied by manipulation of the shape of the pattern pieces used to cut the garment. The tutorial also presents a discussion of the general procedures used in adjusting the paper pattern to accommodate these figure variations.

As these basic concepts are presented, examples are given and questions are asked to ascertain whether these ideas are being understood. The program evaluates the students' response to the study questions and immediately rewards the students for a correct answer. If the students' responses are incorrect, they are given a helpful clue and instructed to try the question again. If students are unable to respond correctly after two tries they are offered the opportunity to go back and review the concept. Each concept which will be judged in the simulation segment is reviewed in the tutorial section of the lesson. Thus the tutorial can be used to provide remedial support when students are unable to apply these concepts in the practice problems of the simulation.

At the end of the tutorial section, a graded Mastery Test is presented which recycles all the previous study questions. To achieve mastery, students must correctly answer 80% of these questions. Since this is a graded test, the students are permitted to go back and review any section before they begin. The student is shown his score on each concept and encouraged to review any section on which his competency score is less than 80%. The Mastery Test must be passed before the

students can go on in this lesson.

INSTRUCTIONS

Following the mastery of conceptual material, the students are presented with a section which teaches them how to use the operations of the simulation section. The students are only presented with this section the first time they go through the lesson. However, they always have the option to return to any part of it.

First, the students are introduced to the options which allow them to move around within the program. As each of these procedural options is explained, the students are instructed to practice working with it. There are four options which the students must understand in order to move around within the program. UNDO THIS STEP allows the students to go back and redo the most recent unfinished operation. UNDO ALL STEPS will erase all work done within a section to be judged. RETURN TO INDEX gives the students the ability to see the section index which then allows the students to move to another part of the Instructions section, return to the Lesson Index or leave the lesson completely. Once all these procedural options are explained and practiced, the students take a quiz which asks them to select the appropriate option to use in a hypothetical situation. Missed questions are recycled until the students have correctly answered 80%.

Once mastery of the procedural options has been achieved, the students move on to explanations and practice with the operational options. Two important points are presented before the students continue. First, the students are shown that a pattern block will appear on an underlay. In the simulation, the student must select the appropriate pattern block for each question. The pattern blocks which are provided with this package are the standard quarter scale sloper for size 12. Next, the students are instructed that the definitions of important terms are available from the GLOSSARY. These terms are underlined when they appear on the computer screen. A message appears telling the students that by pressing the key marked LAB and typing the desired underlined term, the definition will appear. For example, should the definition of the term "block" be requested, the students could read from the response that a block is "a pattern shape with the seam allowances removed."

In order to retain the original styling and drape of the garment, a major concern in pattern alteration is the maintenance of the fabric grain orientation. Therefore, the first operation the students learn is how to DRAW A GUIDELINE on the underlay. By simply indicating the desired location and whether the guideline should be horizontal or vertical, the students can have the program supply them with as many straight guidelines as needed.

Specific locations can be indicated by moving a cross mark (+), called a "cursor", on the computer screen, the cursor can be moved by touching the computer screen or by pressing keys on the keyset which have arrows on them since the cursor will move to a location near the spot touched by the finger, the touch panel can be used for gross placement. Then the "arrowed keys" can be used for refinement. These arrowed keys will cause the cursor to move in the direction of the arrow printed on them. The cursor will be moved one increment per keypress. If the SHIFT key is held down while the arrowed key is pressed, the cursor will move 8 increments/KEY PRESS.

The next operation the students learn is how to MARK A SLASH on the pattern block. Using the arrowed keys, the cursor is moved to the location where the slash will begin. A box labeled ENTER is available on the screen. Once the cursor has been moved to the desired location, the ENTER box is pressed to relay this information to the program. When ENTER is pressed, the cross mark (+) changes into an x. (You know, "x marks the spot".) Following this, the cursor is moved to the location where the slash will end. This placement is indicated to the program as before using the ENTER box. The students can mark a horizontal slash, for lengthening a sleeve; or a diagonal slash for increasing the waistline circumference; two perpendicular slashlines for adjusting the upper back width; or more slashlines for a more involved adjustment such as the change in ease over the abdomen or derriere.

The next operation the students learn permits them to MOVE one or a group of pattern block segments horizontally, vertically or diagonally. This allows them to overlap or spread pattern segments. As the pattern block is slashed into segments, each segment is identified by a letter. The students must indicate which pattern segments are to be moved by pressing lettered boxes on the screen or touching letters on the keyset. If more than one segment is to be moved as a group, these segments are all identified at the same time. Next the students press boxes on the screen or arrowed keys to indicate whether the pattern segments are to be moved to the left, right, up,

down or diagonally..

The next operation further increases the flexibility of the pattern segment moves. The PIVOT operation allows the pattern segment to rotate around one corner of the piece. This operation allows the diagonal slash which was drawn to increase the waistline to be spread at the waistline while the side seam retains the same dimension. Combined with the MOVE operation, the PIVOT operation permits any relocation of block pattern segments that is required in pattern alteration. The students can control the direction of the pattern segment moves by pressing arrowed boxes or keys and they can also control the magnitude of the movement by typing in the desired amount. (The computer's built-in calculator is available, if needed.) The graphic representation on the computer screen interprets the adjustment in quarter scale so the screen display reflects an accurate model of the students' work.

The final operation presented to the students allows them to REFINE the distortion introduced into the original pattern block by the alteration procedures. Using this operation the students can rejoin the discontinuous outline where pieces have been spread. Also they can smooth any "jogs" which have been created along the outer edge of the pattern block by drawing a STRAIGHT LINE between two points. Separated curved edges can be rejoined by copying a designated portion of the edge of a FRENCH CURVE or HIP CURVE graphic. The FRENCH CURVE and HIP CURVE graphics will follow the students' fingers to any location on the screen. They can be rotated 360^{O} and flipped over to perfectly duplicate the utilization of the actual tool. The straight line or curved edge can be used to redraw dart stitching lines. Needed sections can be added and unwanted sections can be deleted (DELETE LINES).

SIMULATION

After completion of the Instruction section, the students are ready to work on their assignment, or they may choose to work out an individual alteration problem. Since the instructor has provided the correct answers for the assigned practice problems the students' progress through these exercises will be monitored and their errors will be caught. Should a student choose to work out an individual alteration adjustment, the available range of pattern blocks will be offered. However, since the possible fitting problems and their subsequent solutions could be infinite, this path is followed without the judgment component. The students working on the assignment are shown the list of practice problems. Those problems which have been completed are identified by the word "done" printed to the left of the problem number. These completed problems are stored and can be recalled to be reviewed or changed. When selecting a new problem to work on, the students are asked to classify the alteration and indicate the pattern block required. If the students respond correctly, they are given the correct answer and asked to go back and select the correct classification or pattern block. Next the students are presented with the appropriate pattern block and proceed to work through their solution to the practice problem. As they complete each step correctly, they are so advised; and should they commit an error which would preclude the correct completion of the problem, they are provided with a helpful hint. When the problem has been correctly completed the students' success is noted. Then they are asked to indicate whether any other pattern pieces will be effected by the changes made in this alteration. If they respond incorrectly, they are directed to select the correct alternative with which the explanation appears. At this point the students are returned to the menu of assigned practice problems to choose the next one to tackle. If the students run out of time or hit a snag, their incomplete problem will be saved until they return to finish. Such problems will be identified by the word "started" alongside the number. Also, students may enter a "help" sequence which will route them back through specific sections of the Instructions or Tutorial Introduction.

The Tutorial portion of the Sewing Pattern Alteration Laboratory Package has been used as a "stand-alone" program to complement classroom instruction. The response of the students has been decidedly favorable. The target date for total completion and refinement of the package is September, 1985. Once this is achieved, expansion to include Flat Pattern Design capabilities is planned.

For further information, or to arrange a demonstration, please contact:

Frances W. Mayhew, 317 Alison Hall or Vickie Gardner, Office of Computer-Based Instruction, University of Delaware, Newark, Delaware 19716.

TEACHING ESTIMATION OF MEASUREMENT THROUGH A MICROCOMPUTER INSTRUCTIONAL GAME

George W. Bright, University of Calgary

ABSTRACT

This study was designed to examine the use of a microcomputer instructional game to teach estimation of length and angle measurement. Previous research indicated that a non-computer version of the game was effective, so the extension to microcomputer format was the primary focus of the experiment. Subjects were 78 pre-service elementary school teachers during spring 1983; data on five subjects were lost. To control for teacher effect, the students of the two instructors constituted separate groups. Within each group, subjects were randomly assigned to play the relevant microcomputer game or another micromputer game which did not deal with measurement. Analysis of the results indicated marginal game effect ($p < .05$ for Group 2 ANOVA of experimental versus control subjects) in estimating angle size but no game effect in estimating lengths of segments. There were different trends for the length estimation data in the two groups, possibly because of an interaction or interference between the classroom instruction and the game instruction. The length distortion on the monitor screens may have also played a role in these results.

This study was designed to determine the effectiveness of a particular microcomputer instructional game in teaching estimation of length and angle measurement. Of primary concern was whether teaching techniques already proven successful in a non-computer version of the game would also prove successful in the computer game.

BACKGROUND

Bright, Harvey, and Wheeler (1982) reported that the game, Geogolf, effectively teaches tenth-grade geometry students to estimate length and angle measurements. In the context of their long-term investigation of non-computer instructional games, this result is not surprising. However, their work leaves open the question of whether the instructional games they used can be adapted to microcomputer format.

The primary difference in the two formats is the manner of presentation of content. Because measurement depends on interpretation of visual information, the monitor screen might present measurement with different effectiveness than a non-computer format; that is, it is not obvious whether a microcomputer version of the game would be as effective as a non-computer version in presenting measurement information in ways that are accurately interpreted by players.

PROCEDURES

Games

Two games were used in this study: Golf Classic (Kraus, 1982a) and Jar Game (Kraus, 1982b). These games deal, respectively, with estimation of angle and length measurements and with

probability. The Jar Game was used as a control, since its instructional objectives were independent of measurement.

Subjects

The subjects for the study were 78 students enrolled in three sections of the mathematics methods course for prospective elementary school teachers at Northern Illinois University in spring 1983. Approximately 90% of the students were female. The three sections were taught by two different people (N = 51 and N = 27), so to control for the teacher effect, the data were analyzed by instructor and were considered as two distinct experiments. Data from five subjects were lost due to incomplete test records.

Treatments

Within each group the subjects were randomly assigned to play one of the two microcomputer games. Each subject played the game twice; each time, the game was played for 20 minutes. The first time the game was played alone, and the second time the game was played with someone else who had also been assigned to play that game. Thus, each subject played her/his assigned game for 40 minutes. Subjects scheduled their own game-playing sessions; disks of the games were placed on reserve in the resource room where the computers were housed. The game playing period lasted five weeks.

It was recognized that the classroom instruction on measurement provided to the two groups might be different, though because of the use of a common final examination for all sections of the course, it was expected that these differences would be minimal. Hence, no measure of the nature of the instruction was made.

Tests

Tests of estimation were adapted from those provided by the publisher of the games. The tests were given to all subjects prior to the beginning of the game playing period and again after all game playing was completed. The pretesting and the posttesting each took a total of fifteen minutes; tests were completed by students as homework assignments.

RESULTS

Descriptive data are presented in Table 1. Further analysis of the data was completed by computing analysis of variance on each of the posttests, with membership in the appropriate game playing group as the classification variable.

There was only one significant F-statistic; that was for the posttest of angle estimation in Group 2 ($F = 4.33$, $P < .05$). Examination of the data in Table 1 indicated that the golf game players seemed to score higher.

Table 1

Descriptive Data

Variable	N	Mean	Std. Dev.
Group 1			
Pretest of length estimation (Maximum score = 22)			
Exp	26	9.1	4.8
Con	22	10.4	4.3
Total	48	9.7	4.5
Posttest of length estimation (Maximum score = 22)			
Exp	26	11.2	5.4
Con	22	9.6	5.4
Total	48	10.5	5.4
Pretest of angle estimation (Maximum score = 20)			
Exp	26	8.0	2.8
Con	22	8.2	3.3
Total	48	8.1	3.0
Posttest of angle estimation (Maximum score = 20)			
Exp	26	8.1	3.0
Con	22	8.2	2.9
Total	48	8.2	2.9

Variable	N	Mean	Std. Dev.
	Group 2		
Pretest of length estimation (Maximum score = 22)			
Exp	14	11.9	4.7
Con	11	10.6	4.8
Total	25	11.4	4.7
Posttest of length (Maximum score = 22)			
Exp	14	9.8	4.4
Con	11	12.2	2.2
Total	25	10.8	3.7
Pretest of angle estimation (Maximum score = 20)			
Exp	14	7.1	3.1
Con	11	7.8	2.9
Total	25	7.4	3.0
Posttest of angle estimation (Maximum score = 20)			
Exp	14	10.5	2.6
Con	11	7.7	4.1
Total	25	9.3	3.5

Note: Exp = experimental group
Con = control group

As a follow-up to these results, regression analysis was applied, with each posttest as the dependent variable and with the corresponding pretest and the appropriate group membership variable as the independent variables. Only one of the regression coefficients on group membership was significant ($p < .05$); that one was in the regression analysis of the posttest of angle estimation for Group 2.

DISCUSSION

The game, Golf Classic, seems to be marginally effective at promoting learning of angle estimation skills. The improvement of the experimental subjects in Group 2 was significant, though there seemed not to be any effect in Group 1.

Of potentially more interest, however, are the patterns of changes in performance on the length estimation test in the two groups. In Group 1 the experimental group improved ($t = 1.92$, $df = 25$, $p < .05$) while the control group declined ($t = -0.56$, $df = 21$, not significant). In Group 2, however, the experimental group declined ($t = -1.35$, $df = 13$, not significant) while the control group improved ($t = 0.94$, $df = 10$, not significant). This suggests that there may have been an interaction between the instruction provided by the game and the instruction provided by the particular instructor. The instructor for Group 2 did spend more time on measurement than the instructor for Group 1. However, there was no way to reconstruct the particular instruction, so investigating the possible interaction between classroom instruction and microcomputer game instruction must be left for future studies.

A few speculations about the possible interaction, however, are in order. First, the gameboard is quite restricted in size on the monitor screen, and the range of values for the length of a shot is limited. Players may be playing the game more as a number game (e.g., find the range of acceptable shots) rather than as a measurement game. Second, the distortion of the monitor screen may interfere with estimation of length. That is, horizontal shots seem to involve a different scale than vertical shots, because of the graphics characteristics of the Apple computer. Students' perceptions about lengths may thus have to be different for the game than for the measurement exercises completed in class. Angles, however, are not distorted as much on the monitor screen. The particular direction of a shot does not interfere with the measurement of the angle. Third, the orientation of the screen in the field of vision is different than for the paper-and-pencil tests used to measure estimation skill. The discrepancy may increase with more paper-and-pencil practice such as likely happened in the instruction given to Group 2.

That there was no substantial effect of the game, especially on length estimation, was somewhat surprising; Bright, Harvey, and Wheeler (1982) did report universal effects. Several things may have contributed to the muting of learning. Perhaps the games were too elementary for the students involved. Perhaps the tests did not measure the appropriate learning outcomes. Perhaps the instruction provided in class was so effective that the additional learning provided by the games could not be measured.

The data of this study suggest that the transition from traditional techniques to computer instructional techniques should not be considered trivial. Much more study of the ways that students learn in computer instruction environments is needed.

REFERENCES

Bright, G. W., Harvey, J. G., & Wheeler, M. M. (1982, April). Cognitive effects of mathematics instructional games at several combinations of instructional and cognitive levels. Paper presented at the Annual Meeting of the National Council of Teachers of Mathematics, Toronto, Canada.

Kraus, W. H. (1982a). Golf classic (software). St. Louis, MO: Milliken Educational Publishing.

Kraus, W. H. (1982b). Jar game (software). St. Louis, MO: Milliken Educational Publishing.

THE ELECTRONIC SPREADSHEET IN MATHEMATICS INSTRUCTION

Samuel W. Spero
Cuyahoga Community College
Cleveland, Ohio 44115

ABSTRACT

Electronic spreadsheet programs such as VISICALC or Supercalc are introduced as calculating tools to be used by students in mathematics course to practice numerical methods. The calculating power of these electronic spreadsheet programs coupled with their ease of use make them ideal for such classroom applications. Furthermore the personal computer technology capable of running these programs is presently available in many mathematics departments.

An example of how the electronic spreadsheet program can be used to solve a cubic equation is presented here as a case study. The solution is developed using the electronic spreadsheet (in this case Supercalc) and some classroom applications are suggested for use in the calculus sequence. A classroom experiment is presently being planned in which this approach will be used.

Introduction

The electronic spreadsheet is a computer program that permits you to use your computer screen as a large sheet of paper and the keyboard as your pencil and calculator. It is a very effective tool for problem solving.

Under such names as VISICALC, SUPERCALC, MULTIPLAN, 1-2-3 and many others, the electronic spreadsheet has been heralded as the spearhead of the personal computer revolution in business. Its ease of use coupled with its computational power have made the electronic spreadsheet invaluable even to the novice computer user.

In mathematics instruction there are many algorithms, especially in the area of numerical methods that we would like our students to practice. Having the students work out the algorithms by hand (even with calculators) is often too time consuming for any but the most trivial examples. To have the students program the algorithms in BASIC, FORTRAN, or PASCAL assumes a level of sophistication in computer programming which many of our students do not as yet possess. To simply used "canned" programs for solving the problems where the students simply add data and equations defeats to a large extent the potential for learning that exists in using the computer.

The electronic spreadsheet program appears to be the ideal compromise. It is like a computer language in that it eliminates the tedious calculation in a procedure, yet it is as easy to use as the student's own paper and pencil. Much of the syntax of a computer language is absent.

In this paper the electronic spreadsheet program (in this case Supercalc) is used to solve a cubic equation using Newton's method. This method is an algorithm often included in the differential calculus as an application of the concept of the derivative. Some suggestions for classroom use of this approach are also suggested.

This presentation is an introduction to an experiment which will be undertaken in the forthcoming school semester in a calculus course in which the students will use an electronic

spreadsheet program to study algorithms introduced within the calculus course.

The Electronic Spreadsheet

(In this paper we will be using the electronic spreadsheet program called SUPERCALC. The concepts presented do apply to virtually all of the major electronic spreadsheet programs with only minor modifications.)

In many cases the electronic spreadsheet comes onto the screen as soon as you activate the disk drive on your computer. The spreadsheet will look something like Figure 1.

The spreadsheet may be thought of as a grid. Columns in the grid are identified by letters of the alphabet while rows are identified by numbers. Spreadsheets like VISICALC or SUPERCALC include 63 columns and 254 rows. The intersection of a row and column, designated by the column letter and the row number, e.g. B24 is the intersection of column B and row 24, is called a cell. All activities on the spreadsheet take place within a cell.

If the spreadsheet can be compared to your paper, the keyboard is your pencil. On the spreadsheet there is one cell which is highlighted.This is called the active cell. When you type on the keyboard and press RETURN the information you have typed is entered into the active cell. You can move the active cell in a number of ways. Most important, however, is the use of the arrow keys which permit you to move the active cell up, down, to the right and to the left. You will note immediately as you practice with these arrow keys that the spreadsheet "scrolls" as you encounter the various boundaries of the spreadsheet as displayed on your screen.

When you printout the spreadsheet, you get exactly what you have typed into the spreadsheet. Thus the format of your output is typed onto the screen as you want it printed out. No PRINT statements are necessary.

Solving the Cubic

To enter alphabetic information you move the active cell to where you want to enter information. type a quote followed by whatever you want entered onto the screen. For example, the title of the program for solving a cubic equation is:

Solving a Cubic Equation of the Form

$$Ax^3 + Bx^2 + Cx + D = 0$$

Starting with a quote, this phrase is typed onto the keyboard (it appears at the bottom of the screen on an editing line). Pressing the RETURN or ENTER key moves the title into the active cell. (Figure 2)

Numbers can be entered directly without quotes. We can type in the coefficients of the cubic equation in this fashion in the following cells:

The coefficient A is in cell B8
The coefficient B is in cell B9
The coefficient C is in cell B10
The coefficient D is in cell B11

The spreadsheet to this point might look like Figure 3.

To solve the cubic equation we use Newton's method which is presented in virtually any elementary calculus text. (For example, CALCULUS WITH ANALYTIC GEOMETRY by Edwin J. Purcell, Appleton-Century Crofts, 1965, page 161.) In this method we can write the following recursive relation for a solution to the cubic.

$$x_{n+1} = x_n - f(x_n)/f'(x_n)$$

If we were to do this by hand (with the help of a calculator) we would set up the solution in a tabular form.

N	x_n	$f(x_n)$	$f'(x_n)$	x_{n+1}

where N indicates the number of the iteration.

In this method we start with some approximation to the solution (we will start by estimating the solution as D, the constant term in the equation). The solution then proceeds by calculating the corresponding value of the function and its derivative. The next approximation to the solution to the cubic is obtained by subtracting the quotient of the function and its derivative (evaluated at the first approximation to the solution) from the first approximation to the solution. This new value of X is then used as the basis for again evaluating f(x) and f'(x) and calculating the next approximation. This process continues until the difference between successive approximations can be ignored.

On the electronic spreadsheet we not only can type in letters and numbers we can also type in a formula. The

variables in the formula are the cell coordinates for a value that will be substituted into the formula.

The first approximation to the solution (X = 50) is typed into cell B24. The formula for the function should be typed into cell C24. In the same fashion that a number is typed in (without a preceding quote) the function is typed in as follows:

```
B8*B24^3 + B9*B24^2 + B10*B24 + B11
```

where multiplication is indicated by the * symbol and exponentiation by the ^ symbol.

As soon as the RETURN key is pressed, the function is evaluated and its value is placed in cell C24. It is important to understand that the formula is still there. It can be seen at the bottom of the screen when the cursor is moved to C24, but the value placed on the screen is always the value of the formula using the currently defined values of the cells included in the formula.

One of the exciting features of the electronic spreadsheet is that if any of the variables in the formula were to be changed the value of the formula itself would change immediately. This can be easily demonstrated by replacing the first approximation to X by some other value or by changing any of the coefficients.

The derivative function can be similarly typed into cell D24 using the following formula:

```
3*B8*B24^2 + 2*B9*B24 + B10
```

and the value for the next approximation to X into cell E24 using the formula:

```
B24 - C24/D24
```

If these have been typed in the screen will appear as in Figure 4.

The value in E24 is now used in B25 as the starting point for a new series of calculations. (A 2 should be typed into A25 to indicate the second iteration.)

While the subsequent iterations can each be typed in, there is available a feature in all of the electronic spreadsheet programs that permits the replication of a formula in successive rows or columns without having to type in the formula itself. The replication can be performed such that those variables within the formula that must remain constant do remain constant while those variables that are relative to the row or column are treated relatively.

All of the columns including column N can be treated with the REPLICATE feature so that very little extra typing beyond this point is required.

In this example we replicated the iteration 10 times. The results appear in Figure 5.

There are many features to the electronic spreadsheet that have not been discussed but which could easily be introduced. For example, it is possible to print out the actual contents of the cells, i.e. the formulas. The formulas contained in the above are shown in Figure 6.

Various types of formatting are also available to the problem solver, as well as many of the functions that characterize the more traditional programming languages. All of these features together with the many other commands can be introduced as needed.

What is important to appreciate is how little background is required to program an algorithm like the above. When this is compared to the knowledge required to program in BASIC or PASCAL, the differences are considerable. Since the goals of mathematics instruction do not include programming as an instructional objective but do include practice in using the algorithm to solve problems, the electronic spreadsheet program represents a potentially useful tool for use by students in the study of mathematics.

Discussion

The remaining two roots of the cubic can be calculated by first using the electronic spreadsheet to factor out the first root using synthetic division. The quadratic equation that remains can be solved using the quadratic formula.

Since students will be using the electronic spreadsheet directly they can take note of imaginary roots without recourse to complicated programming. There is an IF statement available should the student-programmer be up to the challenge and time permitting.

Once the students have programmed the electronic spreadsheet with the algorithm for solving the cubic it is now possible to seriously begin discussing the effect of the various coefficients on the roots. Students can reasonably be expected to study various combinations of coefficients and comment on their effects on the roots, including

at what point the roots become imaginary.

Students can be asked to use the electronic spreadsheet to generate the values of the function so that they can plot it with the roots that they have determined. Some of the electronic spreadsheet programs include either directly or indirectly plotting capability. For example, data from VISICALC can be used with another program called VISIPLOT to prepare graphs. Similarly a new package called SUPERCALC-3 developed by Sorcim for the IBM Personal Computer and others has extensive graphing capabilities designed into the program itself.

In addition to the cubic equation the author has examined a number of applications of the electronic spreadsheet program to instruction in statistics and in linear algebra.

Summary

Many of the professional mathematics teachers' organizations have identified computer programming as a useful tool for reinforcing those concepts of mathematics that involve numerical computations. The popular computer programming languages such as BASIC and PASCAL that are being used to this end represent in themselves disciplines that require an investment of time not normally available in the mathematics classroom. Electronic spreadsheet programs present the same potential for reinforcing mathematics concepts without the substantial investment in the discipline of programming. As such the electronic spreadsheet programs should be investigated as a tool for mathematics instruction.

FIGURES

```
   |  A  ||  B  ||  C  ||  D  ||  E  ||  F  ||  G  |
 1|
 2|
 3|
 4|
 5|
 6|
 7|
 8|
 9|
10|
11|
```

Figure 1

```
   |  A  ||  B  ||  C  ||  D  ||  E  ||  F  ||  G  |
 1|
 2|                                    3     2
 3|Solving a Cubic Equation of the form Ax  + Bx  + Cx + D = 0
 4|=========================================================
 5|
```

Figure 2

```
   |  A  ||  B  ||  C  ||  D  ||  E  ||  F  ||  G  |
 1|
 2|                                    3     2
 3|Solving a Cubic Equation of the form Ax  + Bx  + Cx + D = 0
 4|=========================================================
 5|
 6|The coefficients of the cubic are:
 7|
 8|A =              1
 9|B =          -35.5
10|C =             10
11|D =             50
```

Figure 3

```
   |   A   ||   B   ||   C   ||   D   ||   E   ||   F   |
 1|
 2|                                                  3     2
 3|Solving a Cubic Equation of the form Ax  + Bx  + Cx + D = 0
 4|=========================================================
 5|
 6|The coefficients of the cubic are:
 7|
 8|A =                1
 9|B =            -35.5
10|C =               10
11|D =               50
12|
13|Determining the first root
14|
15|Newton's method:
16|
17|X     = X    -  f(X )/f'(X )
18| n+1     n         n       n
19|
20|
21|N          X          f(X )     f'(X )    X
22|            n           n         n      n+1
23|-----------------------------------------------
24|          1         50 36800.00 3960.000 40.70707
```

Figure 4

```
   |   A   ||   B   ||   C   ||   D   ||   E   |
 1|
 2|                                          3     2
 3|Solving a Cubic Equation of the form Ax  + Bx  + Cx + Dx = 0
 4|=========================================================
 5|
 6|The coefficients of the cubic are:
 7|
 8|A =                1
 9|B =            -35.5
10|C =               10
11|D =               50
12|
13|Determining the first root
14|
15|Newton's method:
16|
17|X     = X    -  f(X )/f'(X )
18| n+1     n         n       n
19|
20|
21|N          X          f(X )     f'(X )    X
22|            n           n         n      n+1
23|-----------------------------------------------
24|          1         50 36800.00 3960.000 40.70707
25|          2 40.70707 9085.528 2090.995 36.36200
26|          3 36.36200 1553.347 1394.883 35.24839
27|          4 35.24839 89.87399 1234.712 35.17560
28|          5 35.17560 .3717947 1224.501 35.17530
29|          6 35.17530 .0000065 1224.459 35.17530
30|          7 35.17530 -2.2e-10 1224.459 35.17530
31|          8 35.17530 3.97e-10 1224.459 35.17530
32|          9 35.17530 -8.6e-11 1224.459 35.17530
33|         10 35.17530 -2.3e-10 1224.459 35.17530
34|
35|The root is          35.17530
```

Figure 5

	C	D	E
21	$f(X_n)$	$f'(X_n)$	X_{n+1}
22			
23			
24	B8*B24^3+B9*B24^2+B10*B24+B11	3*B8*B24^2+2*B9*B24+B10	B24-C24/D24
25	B8*B25^3+B9*B25^2+B10*B25+B11	3*B8*B25^2+2*B9*B25+B10	B25-C25/D25
26	B8*B26^3+B9*B26^2+B10*B26+B11	3*B8*B26^2+2*B9*B26+B10	B26-C26/D26
27	B8*B27^3+B9*B27^2+B10*B27+B11	3*B8*B27^2+2*B9*B27+B10	B27-C27/D27
28	B8*B28^3+B9*B28^2+B10*B28+B11	3*B8*B28^2+2*B9*B28+B10	B28-C28/D28
29	B8*B29^3+B9*B29^2+B10*B29+B11	3*B8*B29^2+2*B9*B29+B10	B29-C29/D29
30	B8*B30^3+B9*B30^2+B10*B30+B11	3*B8*B30^2+2*B9*B30+B10	B30-C30/D30
31	B8*B31^3+B9*B31^2+B10*B31+B11	3*B8*B31^2+2*B9*B31+B10	B31-C31/D31
32	B8*B32^3+B9*B32^2+B10*B32+B11	3*B8*B32^2+2*B9*B32+B10	B32-C32/D32
33	B8*B33^3+B9*B33^2+B10*B33+B11	3*B8*B33^2+2*B9*B33+B10	B33-C33/D33

Figure 6

AN EXAMPLE OF AUTHORING GUIDELINES USED IN DEVELOPING MICRO-COMPUTER BASED TRAINING COURSES

JAMES A. COOK

STANDARD OIL COMPANY OF CALIFORNIA

ABSTRACT

This paper provides a sample of authoring guidelines developed over a period of more than two years. The particular guidelines listed relate to course structure and course coding. They are one result of over 1000 person-hours of authoring experience obtained during the development of five courses utilizing interactive video.

INTRODUCTION

With the development of our first micro-computer based interactive video training course, the need for authoring guidelines quickly became apparent. Guidelines are useful for a number of reasons. They provide a framework for control of the development process. They may be used to promote consistency, quality, and efficiency. They may also serve as a training tool for new personnel.

Our authoring procedures manual now includes guidelines for front-end analysis, video scripting and production, developing student materials, and course evaluation, in addition to the two areas presented in this paper, course structure, and course coding.

These guidelines were designed not as hard and fast rules, but to offer accepted methods to follow unless circumstances dictate otherwise. They are somewhat dependent on the hardware and software in use, the training objective, and the training situation under consideration.

The hardware in this case is an Apple II+ micro-computer with two disk drives, a color monitor, a 1/2 inch video-tape player, and an interface board that enables the Apple to control the videotape player. The software is an authoring system developed and marketed by Bell & Howell called Professional Authoring Software System (PASS).

The training primarily has been technical in nature. The general goal has been to enable operators in a large computer data center to perform their tasks more effectively and efficiently. The training courses must be available 24 hours a day and be designed so that a student can easily work on a course in 20-30 minute blocks of time.

Our authoring procedures manual is open to revision at any time and the guidelines for course structure and coding have been revised at least twice. I have every reason to believe that revisions will occur on a continuing basis, but as of November, 1983, the course structure and course coding guidelines were as follows:

COURSE STRUCTURE

Our standard micro-computer based interactive video courses are developed and organized using the following terms:

```
--- Welcome Unit
--- Menu
--- Modules
    -- Introduction (intro)
    -- Screens
    -- Questions
       - Progress/Review
       - Test
    -- Video Segments
--- Video Sequence Table
--- Workbook
```

For the purposes of this authoring procedures manual these terms are defined as follows:

Welcome Unit: This is a file on the lesson disk containing the first series of informational screens which the student sees when signing on to start or continue the course. It contains the following:

-- Course title
-- Terminal objective(s) of the course
-- Command to initialize the videotape player
-- List of student reference materials
-- Branch to course menu

Menu

The menu is a computer file that is created for every lesson disk and is tailored for each specific lesson disk.

The menu indicates the selections (modules) available in the course and has these characteristics:

-- If the student selects a module on the lesson disk which is currently in the lesson disk drive, he or she is branched to the first unit (file) of that module
-- If the student chooses a module on another disk, he or she is told which disk to insert, and the branch is made to the menu
-- If the student does not make a valid choice, he or she is branched to the menu again

Module

A module is a grouping of files (instructional units) that address one objective. Each file presents a number of display screens.

Each module contains:

-- An introduction unit
-- Instructional files composed of text screens, graphic screens, and question screens
-- A test

A module may also contain one or more video segments.

Intro

This is the first file of each module. It includes a brief statement of the training objective for the module and provides a mini-menu. The student is given three choices:

-- Start the module
-- Try the module test
-- Return to the course menu

Screens

A screen is the display area of the CRT. This term and associated concepts are covered in detail in the Course Coding section of this manual.

Questions

There are two types of questions that are used to monitor student achievement:

-- Progress/review questions
-- Test questions

Progress Review Questions

These questions check the student's understanding of material just covered. They are used to reinforce learning. In general, a student is given one to three chances to answer this type of question correctly. If the correct answer is not entered on a first or second try, a response pointing toward the correct answer is displayed. If all tries have been exhausted without resulting in a correct answer, the student is told the correct answer and given a very brief review (one screen). These questions also prepare the student for the module test.

Progress/Review questions:

-- Are used after every 2 to 8 text or graphic screens
-- Have four answer choices on multiple-choice questions
-- Avoid negatives in question text
-- Require exact matches in answers only when necessary
-- Contain answer judging that allows for misspelling and mistyping where possible
-- Use system commands to limit the number of spaces in an answer
-- Use system commands to lock the keyboard, making it difficult to enter "accidental answers"
-- Include each of the following
  - a positive reply for the entry of a correct answer
  - a "corrective" reply for any unanticipated incorrect

answers
- different helpful replies for anticipated incorrect answers

Test Questions — These questions test the student's mastery of the module objective(s).

The testing unit:

-- Contains an introduction screen that tells the student the total number of questions and how many have to be answered correctly in order to receive credit for completing the module
-- Is composed of questions that the student has only one chance to answer
-- Responds to the student's answer with the word "CORRECT" or "INCORRECT" displayed on the same screen as the question
-- Ends with a screen that tells the student if he or she has passed or failed and what to do next

Video Segment — A video segment is a short (30 sec to 5 min) video display that is used to illustrate concepts that would be difficult to teach with computer text and graphics only.

Video Sequence Table — This is a table on the lesson disk where the beginning and end of each video segment used in the course is stored.

Workbook — The student workbook (in-printed format) contains:

-- Instructions for taking and completing the course
-- A list of the modules presented in the course
-- A list of reference materials
-- Diagrams which can not be reproduced on the microcomputer
-- Any text that the student may need to retain

COURSE CODING

General Course Guidelines — Course coding is accomplished by translating the written course materials into computer executable code. When performing this task there are a number of guidelines to remember about each of the following:

-- Maintenance
-- Documentation
-- Graphics
-- Use of Color
-- Screen formatting and spacing

Maintenance
-- Keep the authoring screens short, try to limit course code to one author screen per student screen
-- Use subroutines when appropriate, subroutines can:
- Keep authoring screens short
- Keep maintenance simple
- Save coding time when the same commands are used repetitively
- Improve consistency
-- Limit the size of any one file to about 15 student screens. This allows for major maintenance changes and is comfortably below our authoring system limits.
-- Keep track of the space ured on the lesson disk. For our authoring system, a maximum of 400 sectors should be used on any new lesson disk. If more than 428 sectors are used, the option of a single disk presentation system is lost.
-- Code a "HELLO" program on each lesson disk so that if it is booted accidentally, the student is given information on how to correct the situation.

Documentation: Document on the written course materials the following:

- Any changes made while entering code
- Module names and file names as they change
- Video segment names

Graphics — Use graphics prudently. While graphics may help alter the pace of a course, they are often time consuming to author. When overused, they can slow the pace of the course and distract the student.

Use of Color

Color is used to increase the attractiveness of the course and hopefully the student's interest and achievement. A few specific guidelines for using color are:

-- Code pleasing color combinations (for example green with white)
-- When using colored text, use only larger characters, smaller characters are very difficult to read
-- Blue or green lines are effective for framing screens
-- Color can also effectively delineate key words of the instructional material

Screen Formatting and Spacing

Some specific guidelines for formatting a screen are:

-- Limit text on screens to a maximum of 14 lines (at least 50% of the screen should be blank space)
-- Limit the number of characters in an authoring line to less than 131
-- Left justify text, if possible
-- Indent when multiple choice answers or lists are very short or line concatenation causes noticeably irregular spacing
-- Do not use the smallest character size except for labels or on menu screens

In addition to the general course guidelines, there are a number of specific guidelines concerning the coding of the following course components:

-- Information/review screens
-- Menu screens
-- Progress/review question screens
-- Test question screens

Information Review Screen

These screens present the instructional material to the student and are made as interesting as possible. To facilitate this goal the following guidelines are useful:

-- Code a pair of green or blue lines at the top and bottom of these screens. (This is a good place to use a subroutine.)
-- When appropriate, use window and erase commands to clear a portion of these screens. This allows graphics and/or information at the top or bottom of a screen to remain if needed and not be redrawn.
-- Avoid blank screens; code in a message to the student explaining what is happening.

Menu Screens

Menus are displayed on only one screen whenever possible and have the following characteristics:

-- Smallest character size is used if space is needed
-- The selections available on each disk are indicated by a frame around them
-- The frame around the modules on the disk in use is a different color (orange)

Progress Review Question Screens

These screens have the following characteristics:

-- The number of tries allowed is indicated on the author screen
-- Lines are not coded at the top and bottom as on information or review screens
-- Each question is numbered
-- Answers to a multiple-choice question are letter choices (i.e, a, b, c, d)
-- Answer judging is coded to give the student the benefit of the doubt
-- Incorrect, unanticipated, and failure responses are specific, when possible
-- When possible, answer responses are displayed on the same screen as the question
-- When commands that restrict the possible input from the keyboard are used, they are turned off immediately when they are no longer needed

Test Question Screens

Test questions are different from the progress/review questions. The student is given only one try at each test question and there are only two possible responses: CORRECT or INCORRECT. These screens have the following

characteristics:

-- The correct response is coded so that it appears on the same screen as the question
-- The correct response contains a quick, bright reinforcing sound
-- The incorrect response also appears on the same screen as the question but contains no sound
-- The number of tries allowed is NOT indicated on the author screen
-- Lines are not coded at the top and bottom as on information or review screens
-- Each question is numbered
-- Answers to multiple-choice questions are letter choices (i.e., a, b, c, d)
-- Answer judging is coded to give the student the benefit of the doubt
-- When commands that restrict the possible input from the keyboard are used, they are turned off immediately when no longer needed

The final course coding guidelines deal with naming conventions for the course files and video segments. These guidelines are very important for course maintenance purposes.

The information about these guidelines is broken down into four sections:

-- Filename conventions
-- Examples
-- Exceptions
-- Video naming conventions

Filename Conventions

The name of a file (instructional unit) always starts with the course name followed by a ":". Next is a three digit module number right justified with a leading zero, if necessary. Most names will end with a three digit unit code beginning with the letter "U" and ending with the unit number right justified with a leading zero, if necessary.

Examples:

Filename	Course Name	Module Number	Unit Number
KOM30:M01U01 =	KOM30,	module 1,	unit 1
KOM20:M01U11 =	KOM20,	module 3,	unit 11
KOM30:M10U01 =	KOM30,	module 10,	unit 1

Exceptions

-- The course welcome unit is given the unit name of "WEL"

Example: KOM30:WEL

-- Each module introduction unit is given the unit name of "U00"

Example: KOM30:M03U00

-- Each module testing unit is given the unit name of "TEST"

Example: KOM30:M03TEST

-- A unit of review questions may be given the unit name of "RQ"

Example: KOM30:M03RQ

-- The course menu is given the unit name of "MENU"

Example: KOM30:MENU

Video Naming Conventions

Video segments are named in the video sequence table. The name always starts with "M" and the module number where the segment is played. The next three characters are "U" and the unit number of the unit where the segment is played. The last two characters are "V" and a number to indicate the location of the segment relative to the other video segments in the same unit.

Example: M01U04V2 is the name of the second video segment in unit 4 of module 1.

CONCLUSION

Many of these guidelines may seem to be fairly basic in nature, but they offer a reasonable foundation for creating training courses with our hardware and software. These guidelines also serve as criteria for evaluating other micro-computer packages. Some of the problems exhibited by these packages include: too much text on a screen, too small a character size for the Apple II+ to display satisfactorily, and screens left blank while the micro-computer goes off to find and load the next file. One can hope that as micro-computer instruction matures, reasonable guidelines will help to improve the quality of training and education courses available.

DOES LEARNING TO PROGRAM IMPROVE CHILDREN'S THINKING SKILLS ?

Erik De Corte

University of Leuven, Belgium

ABSTRACT

In this paper two related problems are treated which are currently the topic of many debates in discussions relating to the cognitive effects of learning to program a computer : (1) Does learning to program improve children's thinking skills and problem-solving abilities ? (2) To what degree is systematic teaching necessary to guide children's learning to program ?

An important argument for teaching computer programming to children is "the contention that learning to program a computer can enhance a student's intellectual functioning" (Coburn e.a., 1982, p. 52). Papert is certainly one of the most prominent advocates of this point of view, which he explains so enthusiasticly in his exciting book Mindstorms (1980). In line with Papert's ideas grand claims have been made : through learning to program children would acquire powerful, general cognitive abilities and skills such as planning, heuristic strategies, and reflectiveness. At the 1983-meeting of the American Educational Research Association held at Montreal in April Pea summarized this viewpoint as follows :

> "Programming is viewed by many of its devotees as a Wheaties of the Mind, a panacee for the ambiguities of everyday cognition. It is alleged that in the demands which programming activities make on the mind - of precision, of problem decomposition and of debugging - programming renders salient the general utility of such cognitive activities in problem-solving efforts, and that such generalizations will be spontaneously made by the programmer to problem spheres above and beyond the microcomputer environment (Pea, 1983, p. 3; see also Pea & Kurland, 1983)".

Coming from a country where the formative value of the classical languages is still a matter of debate, the preceding argument sounds familiar to me, but it also forces me as an instructional psychologist to put forward the following question : what evidence is there available to support the hypothesis that learning to program has generalized positive effects on children's cognitive skills ? Translated in the usual terminology of instructional psychology the problem can be stated as follows : does learning to program result in the acquisition of conceptual knowledge and thinking methods which can be transferred to other content domains ? Several reasons force me to put forward this question.

First of all, the history of the psychology of learning but also more recent studies (e.g. Simon & Hayes, 1976) show that achieving transfer of cognitive skills is not an easy matter, even not in adult learners; transfer certainly has to be pursued explicitly. Moreover, research in cognitive psychology has recently produced robust evidence that problem-solving ability within a given domain of content strongly depends on domain-specific knowledge (Glaser, 1983).

In 1983 I undertook a study trip to the U.S.A. which was focused towards the influence of computer experience on children's cognitive development. I have come to the conclusion that, at present, there is only scarce evidence supporting the transfer hypothesis. However, I must at the same time admit that there is also no counterevidence, and that there are very few good empirical studies relating to this hypothesis. As far as I know, Papert and his associates have only reported anecdotal data to support their position. Their data have been collected mainly in the well-known "Brookline Logo Project" in which 16 sixth graders participated; they all had hands-on experience with the computer varying between 20 and 40 hours (Papert e.a., 1979, Watt, 1979). With respect to the transfer question the following statement is given in the summary of the project :

> "We obviously would like to know how the knowledge gained in the Logo environment is

BEST COPY AVAILABLE

> integrated by the individual student into other activities in school and out of it. We did not have the resources to study this kind of question in great detail and, in any case, are inclined to believe that the time course of the study was too limited for deep effects to show themselves. We did, however, probe the transfer of a few parts of what is learned in LOGO that is particularly close to topics of study in the normal classroom. For example, we shall report below on a simple experiment to show that working with Turtles leads to a measurable improvement in the ability to estimate angles" (Papert e.a., 1979, p. 1.13).

During a symposium on "Computer-based instruction : a future-oriented tutorial" held at the 1983-meeting of the American Educational Research Association the Canadian scholar Guy Groen took a very critical position toward Logo. According to Groen no evaluation studies have been published, simply because no significant effects have been found. Let me add that Groen was very well acquainted with the Brookline Logo Project (see Papert e.a., 1979, p. 1.26), and that he would not expect better results for other computer languages like BASIC or UCSD-Pascal.

The transfer problem was the central issue in a recent study of the Bank Street College of Education in New York (Pea, 1983; Pea & Kurland, 1983). Two classes of 25 children (8-9 year olds, and 11-12 year olds) could have hands-on experience with Logo during one school year; in each class six machines were available. No systematic instruction was given; the researcher followed what he calls "Papert's discovery learning pedagogy". At the end of the school year a series of tests were administered. Analysis of the results showed that in both age groups the experimental class did not display better planning skills than a control class which had no programming experience.

With respect to this study at least two critical remarks can be made :
1) Was the amount of programming time sufficient to obtain significant results ? (Nowadays several experts think that 50 hours of active learning time is an absolute minimum to achieve any significant effects).
2) Would there have been a better chance to achieve transfer effects if, instead of the self-discovery method, a more systematic type of instruction had been applied ?

The state of the art with respect to the transfer problem can be summarized as follows. At present there is no convincing evidence that learning to program results in the acquisition of generalizable and transferable conceptual knowledge and thinking methods. A more extensive review of the literature by Pea & Kurland (1983) leads to the same conclusion. On the other hand the transfer hypothesis can be retained for further study in more systematic and better designed experiments. Only recently some such studies have been started or are in the planning stage. I think esp. of a current investigation by Milojkovic at the Department of Psychology of Stanford University.

Milojkovic is strongly interested in the cognitive and motivational effects on children of learning to program, and also in the most efficient method of teaching programming to children. Hundred fifth graders (10 year olds) participated in his study, divided over five groups. In four groups members of the research team gave instruction during 10 weeks (each week three times 45 minutes) :
- in group 1 children learned Logo;
- in group 2 children learned Basic;
- in group 3 an educational game "Rocky's Boots" was applied (according to some people this game would also stimulate children's programming and thinking skills);
- in group 4 : The productive thinking program : A course in learning to think (Covington e.a., 1979) was used to improve children's thinking skills; these children had no computer experience at all.

Group 5 was a control group in which no instruction was given. In groups 1, 2 and 3 there was one computer available for every two children. In all groups a series of pre- and posttests were administered; and in the experimental groups observational data were also collected.

In Milojkovic's study learning to program was guided somewhat more systematically than Papert's self-discovery strategy advocates. This researcher paid also a lot of attention to the methodological aspects of his study. At present, the results are not yet available. I myself look forward with great interest to the findings of this well-designed piece of research work. However, already now one can ask if 22, 5 hours of active learning time was sufficient hands-on experience to achieve significant effects.

The second question that I want to discuss briefly, has already shown up several times in what preceeds - namely, to what degree is systematic instruction necessary to guide the learning process with respect to programming.

In Mindstorms (1980) Papert advocates the self-discovery strategy for learning to program with Logo. According to him, learning Logo should take place in a spontaneous and natural fashion, "analogous to how a child learns to speak" (Papert, 1980, p. 27). He calls it learning without curriculum, its counterpart being teaching without curriculum (Papert, 1980, p. 31). This conception is based on the work of Piaget, who himself gave very few statements on teaching. Furthermore, Papert's view of learning gives rise to a fundamental criticism of the traditional teaching methods and of the existing school system as a whole.

> "I see the classroom as an artificial and inefficient learning environment that society has been forced to invent because its informal environments fail in certain essential learning domains, such as writing or grammar or school math. I believe that the computer presence will enable us to modify the learning

environment outside the classrooms that much if not all the knowledge schools presently try to teach with such pain and expense and such limited success will be learned, as the child learns to talk, painlessly, succesfully, and without organized instruction. This obviously implies that schools as we know them today will have no place in the future" (Papert, 1980, p. 8-9).

With respect to this point of view I also have some critical remarks. However, let me state beforehand that I agree strongly with Papert's position that a programming language can only be stimulating, motivating, and instructive if the children have the opportunity to engage actively in the programming environment. I state this explicitly because I certainly do not agree with certain practical applications which have already shown up, and which can be characterized as nullification by assimilation, a well-known phenomenon in the history of education : innovations are fitted into and adapted to the prevailing educational practice, loosing thereby their most essential characteristics. In the April 1983 Logo-issue of Classroom Computer News, Kelman (1983, p. 7) draws our attention to this phenomenon :

"Logo curriculum guides come replete with behavioral objectives, Logo workbooks, and Logo multiple-choice tests. Nothing could be more antithetical to the philosophy underlying Logo's development".

On the other hand, I do not think that a more or less extreme form of the self-discovery method is the appropriate teaching device for programming. Decades of research in educational psychology support my scepticism in this respect. This research shows that there is no one and only superior teaching method, but that the combined and alternating application of a variety of methods offers better perspectives for successful teaching. Leron's (1983) extensive experience with children's Logo learning and the already mentioned Bank Street study (Pea, 1983; Pea & Kurland, 1983) suggest that this probably holds also with respect to programming. In this last investigation with two classes of 25 children (8-9 year olds, and 11-12 year olds) the discovery method was not very effective not only with respect to the acquisition of problem-solving skills in general, but even for the mastery of different programming skills.

It seems to me that the key question is not : self discovery or not, direct instruction or not ? The main problem is rather to give the learners the appropriate help and guidance at the right moment. Moore (1983, p. 14) seems to share this position. She writes :

"Knowing when to intervene and when not to seems to be the secret of artful Logo teaching".

An important general criterion in this respect is probably that one should not take the task off the learner's hands, so that (s)he remains in control. In deciding what kind and how much help to give in a specific teaching-learning situation, one should take into account the nature of the learning task and certain characteristics of the learners, such as their prior learning, level of ability, and level of motivation. No data being available, I would speculate that a discovery-type method can be successful for highly motivated, high (to average) ability learners; with less motivated, low ability pupils a more direct and sequenced form of teaching will probably be more effective.

With respect to both related topics discussed in this paper - namely, can learning to program promote children's general problem-solving abilities, and to what degree does learning to program require systematic teaching ? - further research is necessary before more definitive and more specific answers can be given. It is therefore high time to undertake well-designed, longitudinal studies in the near future. This position is not only inspired by my theoretical interests as a researcher but also by my concern for educational practice. On the one hand I believe that the microcomputer is a potentially powerful tool which in the future might contribute significantly to the improvement of education. However, I am concerned when I see mostly poor practice of computer use in schools today. Recently I had the opportunity to talk to an expert in this domain, Alfred Bork of the University of California at Irvine; he strongly supported my concern stating that at present computers probably do more harm than good in American schools (1). It is my bias as an investigator that good research can build up a body of knowledge which can help us in achieving a well-guided, appropriate, and thoughtful implementation of computers in education, a conditio sine qua non to guarantee that those machines will affect educational practice in a significant and favorable way.

Note

(1) Personal communication by Alfred Bork, August 1983 (see also Bork, 1983a, 1983b).

References

Bork, A., Computers and information technology as a learning aid. Paper presented at a conference held at the Centrum voor Onderwijs en Informatietechnologie, Technische Hogeschool Twente, Enschede, Holland, June 1983. (1983a)

Bork, A., Computers and the future : education. Paper presented at the Computer-assisted learning conference 1983, held at the University of Bristol, May 1983. (1983b)

Coburn, P., Kelman, P. Roberts, N., Snyder, T.F.F., Watt, D.H. & Weiner, C., Practical guide to computers in education. Menlo Park, Cal. : Addison-Wesley, 1982.

Covington, M.V., Crutchfield, R.S., Davies, L. &

Olton, R.M., Jr., The productive thinking program : A course in learning to think. Columbus, Ohio : Ch. E. Merrill, 1974.

Glaser, R., Education and thinking : the role of knowledge. Pittsburgh : Learning Research and Development Center, University of Pittsburgh, 1983.

Groen, G., Paper presented in a symposium "Computer-based instruction : a future-oriented tutorial", at the Annual Meeting of the American Educational Research Association, Montreal, Canada, April 1983.

Kelman, P., Seymour, has your dream come true ? Classroom Computer News, 1983, No. 5, 7.

Leron, U., Some problems in children's Logo Learning. In Proceedings of the seventh International Conference for the Psychology of Mathematics Education. Jerusalem : Weizmann Institute, 1983.

Moore, M.J., The art of teaching LOGO or when and when not to bother the learner. Hands-on. Microcomputers in education : innovations and issues, 1983, 6, 12-14.

Papert, S., Mindstorms. Children, computers, and powerful ideas. New York : Basic Books, Inc., 1980.

Papert, S., Watt, D., diSessa, A. & Weir, S., Final report of the Brookline LOGO Project, Part II : Project summary and data analysis. (LOGO Memo, No. 53). Cambridge, Mass. : M.I.T. Artificial Intelligence Laboratory, 1979.

Pea, R.D. & Kurland, M., On the cognitive and educational benefits of teaching children programming : a critical look. New Ideas in Psychology, 1983, 1, in press.

Pea, R.D., Logo programming and problem solving. Paper presented at a symposium entitled : "Chameleon in the classroom : Developing roles for computers", Annual Meeting of the American Educational Research Association, Montreal, Canada, April 1983. (Also in : Technical Report No. 22. New York : Center for Children and Technology, Bank Street College of Education.)

Simon, H.A. & Hayes, J.R., The understanding process : Problem isomorphs. Cognitive Psychology, 1976, 8, 165-190.

Watt, D., Final report of the Brookline LOGO Project. Part III : Profiles of individual student's work. (LOGO Memo, No. 54). Cambridge, Mass. : M.I.T. Artificial Intelligence Laboratory, 1979.

THE CONVERSION OF PLATO® COURSEWARE TO THE APPLE® MICROCOMPUTER

LOUISA FRANK AND LYNN SMITH
UNIVERSITY OF DELAWARE

ABSTRACT

Many steps of the development process must be reconsidered in any conversion of courseware to a different system. Although content and instructional design need not be changed, redesign of screen displays and reprogramming pose problems similar to those frequently encountered in original development. Two conversion projects are being carried out at the University of Delaware's Office of Computer-Based Instruction (OCBI). Problems that arose during conversion as well as tools and solutions that were developed to overcome the problems are described.

INTRODUCTION

Three main areas must be considered in developing computer-based instructional (CBI) materials:

1. content and instructional design
2. screen display design and interaction
3. programming

In addition, review and revision of each of these areas is important.

Converting existing CBI materials from one computer to another does not require reconsideration of the first area, content and instructional design. However the second and third, screen display design and programming must be reexamined. The amount of attention to be given each area is determined by the differences between the display size and programming language of the two computers. Review and revision activities should concentrate on the changes necessary to adapt to the new computer, ensuring that the lesson flow and coherence are retained.

At the University of Delaware's Office of Computer-Based Instruction (OCBI), two projects have been funded for conversion of PLATO courseware to the Apple microcomputer in the areas of Latin and mathematics.

One set of lessons undergoing conversion consists of five lessons, originally developed under a National Science Foundation* grant as PLATO courseware, dealing with applied mathematics. The lessons, each four to six hours in length, are designed as enrichment materials for motivated secondary students. They were initally developed on a mainframe to facilitate data collection for revision and evaluation purposes. However it was evident that conversion to microcomputer was necessary to reach secondary students.

To accomplish this conversion, a team of one and one half programmer/analysts, divided between design experts and programming experts; a part-time designer; and three part-time programmers was assembled.

Because of the length of the original PLATO lessons that first design task was to split each lesson into

PLATO® is a registered trademark of Control Data Corporation.

Apple® is a registered trademark of Apple Computer, Inc.

*This project is supported in part by National Science Foundation Grant Number SED 8025787. Any opinions, findings, and conclusions or recommendations expressed by this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

subtopics to be contained on individual disks. Some of the lessons had natural break points, while others were not as easily subdivided and had to be rewritten. It was found that a total of fifteen disks would be necessary. Once this decision was made design of the individual disks was begun.

REDESIGNING SCREEN DISPLAYS: SOME USEFUL UTILITIES

A first step was the redesign of the screen displays since the configuration of the Apple differs from that of a PLATO terminal. A decision was made to use two utility lessons created earlier for the Latin conversion project, started in September of 1982. Like the Mathematics project, the Latin project involved converting several PLATO lessons to the Apple microcomputer. When the lessons were ready to be programmed, it was realized that valuable time would be spent on the tedious task of entering text and graphics statements. This time could be better utilized for programming more difficult and demanding routines; therefore, "Micro Script Converter" (MSC) was created.

A utility lesson, "Micro Script Converter" (MSC) is written in the TUTOR programming language; it allows text entry and simulates the appearance of the text on a microcomputer screen of 40 or 80 characters by 24 lines, as illustrated in figure 1. The Latin courseware, like most lessons, had to be rescripted to accommodate the smaller size of a microcomputer screen. Since the lessons contained a large amount of text and graphics, creating the script using MSC - which features quick scripting and placement of text and graphics for microcomputers - eliminated the time consuming task of typing text on a microcomputer.

For conversion of the mathematics lessons, transfering the design from a 64 character by 30 line display proved somewhat easier than was anticipated. One factor which helped a great deal was the significant border of the Apple screen, lacking in the 64 by 30 character screen. This permitted utilization of all 40 characters in width, instead of having to leave several blank character spaces to form a border. In addition, a BASIC enhancement was used for programming which allowed upper and lower case. Thus more text could be put on the screen and still be clear and readable. Displays were often translated intact, as seen in figures 2 and 3. In other situations a simple split of one large screen into two smaller ones was effective, as illustrated by figures 4, 5, and 6. Substantial changes necessary in only a few cases.

In MSC a typical display shows a box representing the dot by dot screen size of a microcomputer (5 x 7 dot characters in a 280 x 192 dot screen size.) Screen displays can be designed by

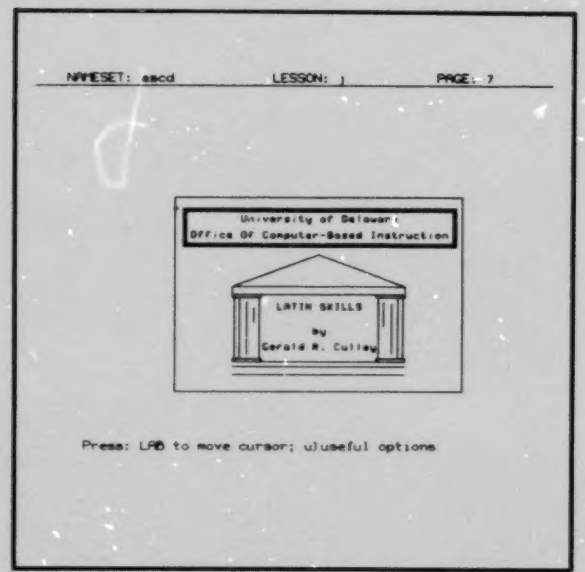


Figure 1. Screen display from "Micro Script Converter".

The store gives you the following prices for all ten sizes corresponding to your shrubs. You must choose three sizes to be manufactured, and and decide how many of each you need.
Cover Size Number (x)
Cost (C_x)
1 C_1 = $ 1
2 C_2 = $ 4
3 C_3 = $ 5
4 C_4 = $ 7
5 C_5 = $ 8
6 C_6 = $ 12
7 C_7 = $ 13
8 C_8 = $ 18
9 C_9 = $ 19
10 C_{10} = $ 21
The objective is to choose the three sizes which enable you to cover all 10 shrubs as inexpensively as possible.
Press: NEXT to continue
BACK to review

Figure 2. Screen display from PLATO version of "Shrub Covering Problem"

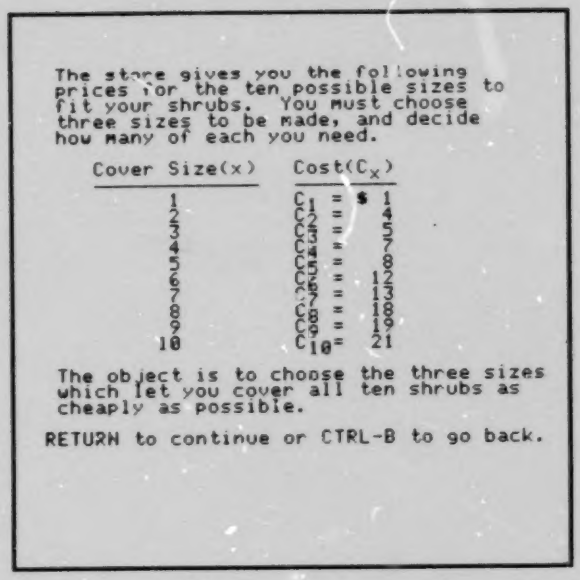

The store gives you the following prices for the ten possible sizes to fit your shrubs. You must choose three sizes to be made, and decide how many of each you need.
Cover Size(x)
Cost(C_x)
1 = $ 1
2 = 4
3 = 5
4 = 7
5 = 8
6 = 12
7 = 13
8 = 18
9 = 19
10 = 21
The object is to choose the three sizes which let you cover all ten shrubs as cheaply as possible.
RETURN to continue or CTRL-B to go back.

Figure 3. Screen display from Apple version of "Shrub Covering Problem"

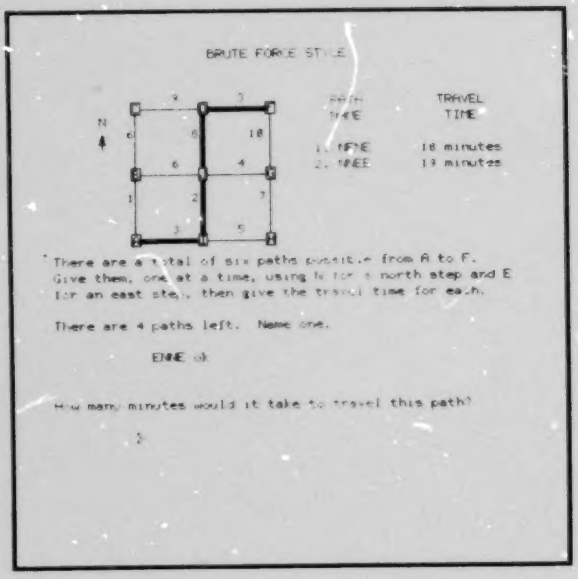

BRUTE FORCE STYLE
PATH
TRAVEL TIME
1. NENE
18 minutes
2. NNEE
17 minutes
There are a total of six paths possible from A to F. Give them, one at a time, using N for a north step and E for an east step, then give the travel time for each.
There are 4 paths left. Name one.
ENNE ok
How many minutes would it take to travel this path?

Figure 4. Screen display from PLATO version of "Shortest Path Problems"

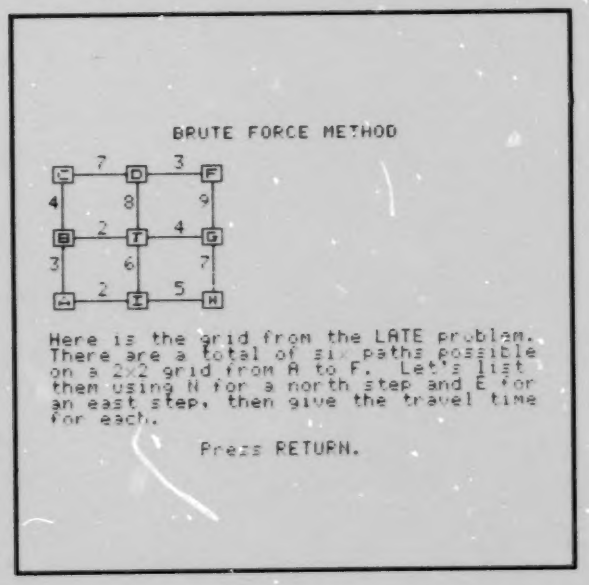

BRUTE FORCE METHOD
Here is the grid from the LATE problem. There are a total of six paths possible on a 2x2 grid from A to F. Let's list them using N for a north step and E for an east step, then give the travel time for each.
Press RETURN.

Figure 5. First screen display from Apple version of "Shortest Path Problem"

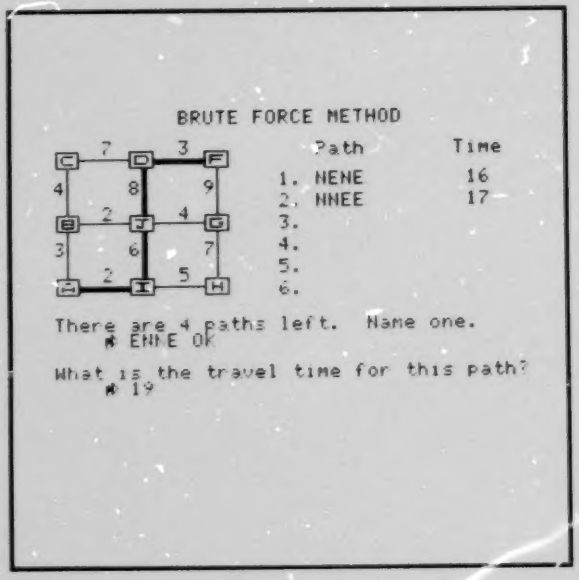


Figure 6. Second screen display from Apple version of "Shortest Path Problem"

choosing a location and entering the text. The user can move the text to another location, edit the text or remove it entirely. The user also controls whether new text being entered plots before or after existing text. Figure 7 shows the various options available to the user.

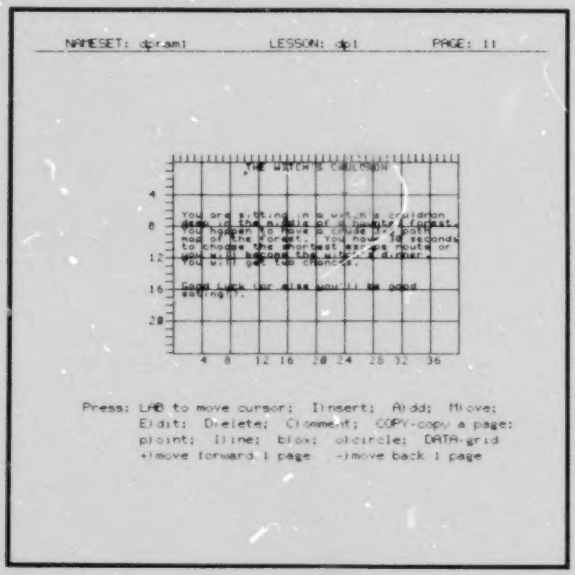


Figure 7. Screen display from "Micro Script Converter" showing user options

MSC does not limit the user to just text entry; graphics can also be created, including boxes, points, lines, and circles. Once drawn, they can be saved along with the text in a datafile attached to the lesson. When the user is finished entering text or graphics, the information saved in the datafile can be transfered to a microcomputer.

A second utility lesson, also programmed in TUTOR, translates the data in the datafile to the desired language. Since the Latin project used APPLESOFT II BASIC, the second utility was set up to translate data into APPLESOFT II BASIC code. It reads the datafile and writes the appropriate code, numbering all the line; in addition it can compensate for a high resolution screen driver and allows for the use of a BASIC code enhancement program. The utility is currently set up to translate data to APPLESOFT II BASIC, but future plans include the addition of other forms of BASIC, as well as Pascal.

Currently, to transfer material from a PLATO terminal to the Apple, PLATO is called up through the ASCII port. The code generation utility is then executed, saving the data in a data capture buffer. Subsequently this data is transferred to a text file and a utility on Apple is run to "clean up" the text file, removing all unwanted characters, such as carriage returns, that PLATO automatically inserts when transferring the material through the telephone line. When this is done, the material exists as real, runnable APPLESOFT II BASIC code.

The simplicity of using MSC results from the fact that it requires no prior knowledge of a programming language. It can be used for converting lessons from one computer to another or for initial design of a lesson intended for a microcomputer. While a user is entering text and graphics, a programmer can be busy writing the other quantity and size of text and graphics in a lesson, valuable time can be saved. The use of MSC saved the Latin project approximately two to three months of programming time. Since then, other lessons targeted for a microcomputer have been scripted and coded through the use of MSC and the code generating utility.

The designs for all fifteen mathematics disks were completed in approximately four months. The process included designing the screen displays using MSC; making a script or storyboard with screen copies from the mainframe design utility displays; and adding directions such as those regarding branching or answer judging for the programmer, who was then given a copy of this script to work from.

REPROGRAMMING SOME USEFUL TOOLS

While the designers were working on splitting the original mathematics lessons into disks, the programmers were organizing for their tasks.

In beginning this conversion an important decision was made to adopt an enhancement designed for CBI as a programming tool. EnBASIC®, the enhancement chosen, offers many of the features that are available in the TUTOR programming language. Although only two of the programmers who, at various times, worked on the project were at all familiar with TUTOR, all learned to use this enhancement without difficulty, and found it made accomplishment of the task much easier than it would have been using

EnBASIC® is a trademark of Compress, a division of Science Books International Inc.

APPLESOFT II BASIC alone.

After studing EnBASIC, the programmers noted several additional subroutines which would be useful and set about creating them. While the first disks were being designed, the programmers created a "template disk" containing EnBASIC, some documentation utilities, and several subroutines which were common to all of the lessons. For example, one subroutine made it possible to back up through the lesson, while a second accessed the APPLESOFT II BASIC expression evaluator, allowing students to enter mathematical expressions and have them evaluated before being judged. Others included a section on "How to Use This Lesson," a circle drawing routine, and routines for working on the disk. Once ready the design could be transferred to a copy of the template disk, and actual programming could begin.

One problem with EnBASIC is that there is only 14 K of memory in a 48 K machine available for the program. This was handled by indexing the lesson so that each section comprised one disk access. In some cases it was necessary to sub-index to attain this goal. The index was kept resident at all times to avoid annoying frequent disk accesses.

Another problem was that EnBASIC, while allowing for negative feedback, did not allow for multiple layers of negative feedback. That is, it did not allow for different responses based on the number of times a student answered incorrectly. Programmers devoted many hours to resolving this problem.

Once the programming was complete, changes and compromises made during the conversion process were reviewed. As a result a few revisions were made, and the lessons were readied for dissemination.

CONCLUSION

The University of Delaware has requested federal funds to develop a second set of lessons on other mathematical topics. It is anticipated that these would be developed directly for microcomputers, rather than initially on mainframe, and that the MSC will be used to design the displays

The process of converting mainframe courseware to microcomputer is an abbreviated version of the initial development process. Redesign of screen displays and reprogramming are the major steps in this process, will review and revision activities concentrating mainly on changes necessitated by differing hardware. "Micro Script Converter" is a mainframe utility which greatly accelerates the redesign of screen displays.

(The code generating utility was written by Graham Oberem of Rhodes University while on sabatical leave at the University of Delaware.)

Many of the disks produced by the two projects described are available through the University of Delaware.

EVALUATING CLASSROOM MICROCOMPUTER COURSEWARE

Paul Geisert and Mynga Futrell

University of Wisconsin-Oshkosh

ABSTRACT

Courseware evaluation generally includes the reviewing of program features, but typically fails to assess actual instructional outcomes achieved when the courseware is employed in the classroom. This paper presents a two-phase process for evaluating instructional software. Phase one involves evaluating the design features of both the CAI and the CMI components of the courseware. The paper provides a comprehensive listing of criteria of sound design which can be considered indicators of the instructional value of the program. Phase two of the evaluation consists of conducting an empirical tryout of the courseware to measure its actual instructional effectiveness.

INTRODUCTION

Instructional evaluation involves making professional decisions concerning some instructional product or process. We will present an evaluation plan for making decisions about computer software that has instructional intent. Some aspects of the evaluation plan may be useful for evaluating other forms of classroom courseware (drill-and-practice, games, etc.), but the plan as presented focuses on courseware aimed at promoting student learning.

The plan can be used for making decisions about commercial courseware as well as for materials under development. While the evaluation criteria are essentially the same, the strategies for assessing the two forms of courseware are different. Materials under development are available for modification during every phase of their evolution from idea through classroom use. One can use a formative evaluation process and make decisions to improve the product during development. Most commercial courseware cannot be changed, and one will be using a summative evaluation process to make acceptance or rejection decisions.

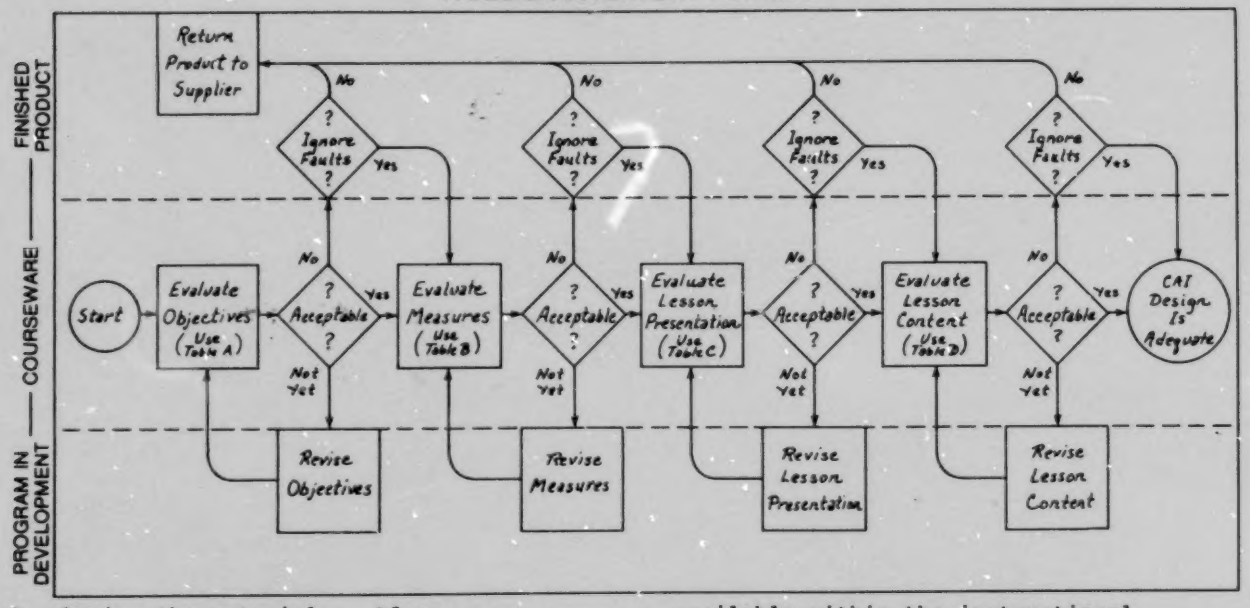
Although this paper will address the reader as if he or she were an individual conducting the evaluation, this strategem is employed to aid in explaining the evaluation model, and is not meant to reflect or define participants in the evaluation process. The number and nature of evaluators can be seen to vary with the importance accorded the courseware, and may vary as well during the evaluation so that appropriate expertise can be brought to bear on the various considerations.

The evaluation plan consists of two major phases. First, a design inventory is conducted to assess the adequacy of the instructional (CAI) and management (CMI) components. Second, an empirical tryout of the materials with students is conducted. This latter step, to ascertain the program's actual contribution to student learning, is essential in determining the worth of a piece of courseware. Sadly, this aspect is seldom systematically examined in typical software evaluation schemes.

EVALUATING THE DESIGN OF CAI

The flow diagram in Figure 1 outlines the logic of the CAI assessment conducted during the design inventory phase of the evaluation. If you are evaluating your own courseware, you will utilize the lower two thirds of the diagram to perform the evaluation while you are

FIGURE 1
A DESIGN INVENTORY FOR CAI



developing the materials. If you are evaluating someone else's courseware (commercial or another teacher's), you will use the middle and top third to evaluate a finished product.

Evaluating Program Objectives. Instructional objectives are the heart of a well-developed instructional computer program. If the objectives are weak, all other program components will probably be weak. Items to be considered when evaluating instructional objectives are listed in Table A. You will need to select which questions are pertinent to the program you are evaluating, and then answer each pertinent question, YES or NO. After completing the questioning process, you must decide whether or not to accept the objectives as they are. The decision will not be easy to make, since it requires that you collapse your answers to the evaluation questions into a single decision.

If the program objectives are acceptable, this step in the design inventory phase is finished and you move on to the next step--evaluating measures. If, however, you view the objectives as inadequate, you will want to improve the program by revising them, which you can do if evaluating your own materials. Of course, if you find that a commercial program's objectives are unacceptable, you will need to decide whether to reject the program on that basis, or whether other considerations (e.g., additional program features present or remedies available within the instructional setting) might cause you to ignore its faults. The first decision results in your sending the material back to the producer; the second will send you further in the evaluation.

TABLE A

Evaluation of Objectives

* Does the program have goals and objectives stated either in the courseware itself or in a teacher's guide?
* Are the objectives stated in behavioral terms?
* Do the objectives state the conditions under which the students will perform?
* Are the criteria for successful performance stated?
* Are the objectives arranged in a progression from simple to complex?
* Do the objectives have educational value?
* Are the objectives appropriate to the students' ages, backgrounds, and characteristics?
* Can the objectives be integrated with previous student experiences?
* Are the objectives generalizable to an appropriate range of situations?
* Are the scope and depth of the objectives reasonable?

Evaluating Measures. Given that the program's instructional purpose is valuable and well-defined, the adequacy of its testing component should be examined next. The process is the same as the one used in evaluating objectives, that is, first considering the relevant questions and answering them YES or NO, and then collapsing your considerations into a single decision on the acceptability of this aspect of the courseware. Items to consider when evaluating measures of student learning are provided in Table B.

TABLE B

Evaluation of Measures

* Does the program have test items that measure student performance?
* Are the test items referenced directly to the objectives of the program?
* Is there a sufficient number of test items for each objective?
* Is a sufficient number of tests and retests provided?
* Is an adequate performance level expected?
* Are the test items technically correct, that is, written clearly, unambiguously, etc.?

Evaluating Lesson Presentation. If the courseware has passed to this point or has been revised to an adequate condition, it is now evaluated on lesson presentation. Do not confuse lesson presentation with lesson content. Here you will give consideration to whether or not design rules for effective instructional processes have been followed in the development of the lessons, and whether these processes are implemented in an appropriate and efficient manner.

With commercial materials, it is often possible to evaluate CAI objectives and tests from a teacher's guide or other information accompanying the computer program. It is rarely possible to do this with lesson presentation. The program will probably need to be reviewed on-line, with you acting "as a student." The questions in Table C should be addressed.

Evaluating Lesson Content. A number of criteria fall under the concept of "content" and should be considered. This does not make for an easy evaluation task, since you will be faced with the problem of having both good and bad aspects in every program. Any one factor might be sufficient to warrant a reject

TABLE C

Evaluation of Lesson Presentation

Instructional Processes

* Does the computer get the students' attention?
* Does it present the purpose of the lesson to the students?
* Does the computer pretest the studerts to determine if they can do the objectives for the lesson?
* Does it remind the students what they have learned in the past and need to know for this lesson?
* Is the lesson presented appropriately for the type of learning, that is, for discrimination, concept learning, and so on?
* Does it provide examples of the expected student performance for the objective?
* Does it provide example helps?
* Does it provide practice items on the objective?
* Does it provide practice helps?
* Does it provide practice feedback?
* Does it provide a posttest to determine if the students have been successful on the objective?
* Does it provide for retention and transfer?

Presentation Features

* Are instructions to students clear and sufficient?
* Does the program have a clear and informative menu structure?
* Is lesson presentation in the form of a tutorial rather than a linear presentation?
* Is the presentation clear and logical?
* Does the presentation progress in appropriate increments?
* Is the program free of errors in grammar, spelling, hyphenation, and punctuation?
* Is the reading at an appropriate level for the students?
* Is the tone of address of program output appropriate?
* Are information displays well organized?
* Are graphics/color/sound used to support the presentation when needed?
* Are cuing and prompting properly employed?
* Is output consistent with input request?
* Is feedback effectively employed?

Table C, continued

* Does the presentation motivate the students to continue their work?
* Do the students have the option of controlling the rate of presentation of the lesson?
* Do the students have the option of controlling the sequence of presentation of the lesson?
* Can students review previous screens and exit to menus at will?
* Will the students learn if they faithfully attempt to complete the instruction as presented?
* Is the instruction efficient, that is, not redundant, boring, or time-consuming?
* Can the system diagnose student performance and branch the students to appropriate instruction?
* Does the program use the concept of "not-passed-yet" rather than "failure" when students have not yet accomplished a task?

TABLE D

Evaluation of Lesson Content

* Does the content support the program objectives?
* Is the content accurate?
* Is the content free of race, ethnic, sex, or other stereotypes?
* Is the content presented objectively and without bias?
* Does the content have educational value?
* Is the content appropriate to the students' backgrounds and interests?
* Is the content sufficient to insure that students can reach the objective if the lesson is studied?
* Is the content at the appropiate level for the students?

decision for commercial materials. For example, if you have a high value associated with teaching content that is free of stereotypes, any stereotyping in the program might be sufficient reason not to purchase it. In another situation, the program might be purchased and labeled "sex (or other) stereotypes present" and then used as a point of discussion.

Some of the considerations listed in Table D will not be pertinent to a given piece of CAI courseware. For example, you may not plan that the courseware be the sole instruction your students receive on a given lesson. You may be willing to supplement the CAI instruction yourself, and therefore would not feel it necessary that the computer lesson contain all the instruction your students would be likely to need in order to achieve the lesson objective.

If an evaluation item in Table D is important, however, then the program in question should be evaluated closely on the criterion. Failure to develop one aspect of a program may not necessarily rule out its purchase, but it certainly does lower the program's overall value. You should of course revise any program you are developing to meet important content criteria.

EVALUATING THE DESIGN OF CMI

The management function within a piece of courseware being evaluated may be an aspect of CAI, or it may stand on its own as a management program. Either way, the computer management entity should follow sound design principles. The process for conducting an inventory of the CMI design aspects of a given computer program is outlined in Figure 2. It is much like the procedure for the CAI inventory already described. You should assess the courseware on each criterion in Tables E and F having significance with respect to the instructional setting within which the program is to be utilized.

If you decide that the concept in an evaluation question is important to the materials you are evaluating, then you must interact sufficiently with the computer program to ascertain whether or not the application of the idea within the courseware is indeed adequate.

Because of the difficulty of the programming task, most courseware having a significant CMI component is a commercial product resulting from a rather large development project. The CMI program you are evaluating might be a completely closed document providing only what the developer has to offer (CMI within a reading program, for example), or, at the other pole, it might be essentially an authoring system, completely open to the user's (in this case, the teacher's) additions. The nature of the program will have to be taken into consideration when each of the questions in the tables is answered. Whatever the program's nature, however, you must try it out on the computer to determine its organization and the ease with which a user can move through it.

As you can see from Figure 2, there are two aspects to assess when evaluating CMI. The aspect of preparing the master diskette should be examined, as should the record keeping capabilities of the system.

FIGURE 2
A DESIGN INVENTORY FOR CMI

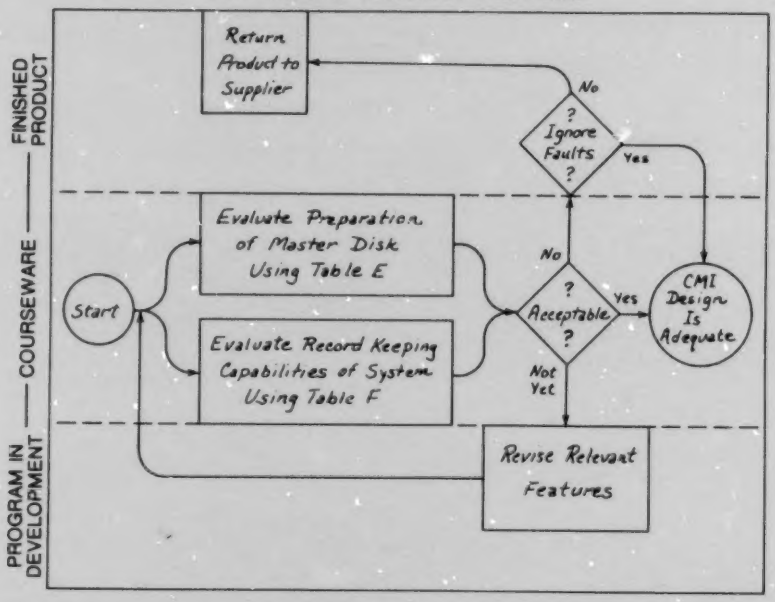


TABLE E

Evaluation of Master Disk Preparation

* Is a manual for using the program provided?
* Does the program have menus?
* Are the menus clear and easy to read and follow?
* Does the teacher have a password?
* Can students be assigned passwords?
* Are provisions made for a sufficient number of students?
* Can students be placed in as many groups as needed?
* Can student groups be named?
* Can students be easily added or deleted from a group?
* Can the teacher enter performance objectives?
* Can the teacher enter test items?
* Can standards be set for performance?
* Can instructional materials be linked to objectives, for prescription purposes?
* Can the teacher opt to use or not use the management system?
* Can the disk be duplicated as often as needed by the teacher?

TABLE F

Evaluation of System Recordkeeping Capabilities

* Is the disk protected so that students cannot see other students' records?
* Are the records of the system easy to locate?
* Are the records easy to read?
* Are the records available by individual students?
* Are the records available by class, for a given objective?
* Are the records available by objective, for a given class?
* Do the records tell when student has attempted an objective?
* Do the records tell how many times a student has attempted an objective?
* Do the records provide information across a span of units and lessons?
* Does the CMI system have an optional print capability for student records?
* Does the system have an optional print capability for producing printed tests for specific objectives?
* Does the system have an optional print capability for listing the objectives of the program?
* Can the system print parent reports on what the students have learned using the program?

OTHER CONSIDERATIONS

Table G provides a listing of a few factors not specifically associated with courseware design which therefore do not fit conveniently within the areas considered. These additional concepts relate primarily to aspects of program acquisition and use. You will want to take some of these factors into consideration when deciding whether or not to purchase a piece of commercial courseware.

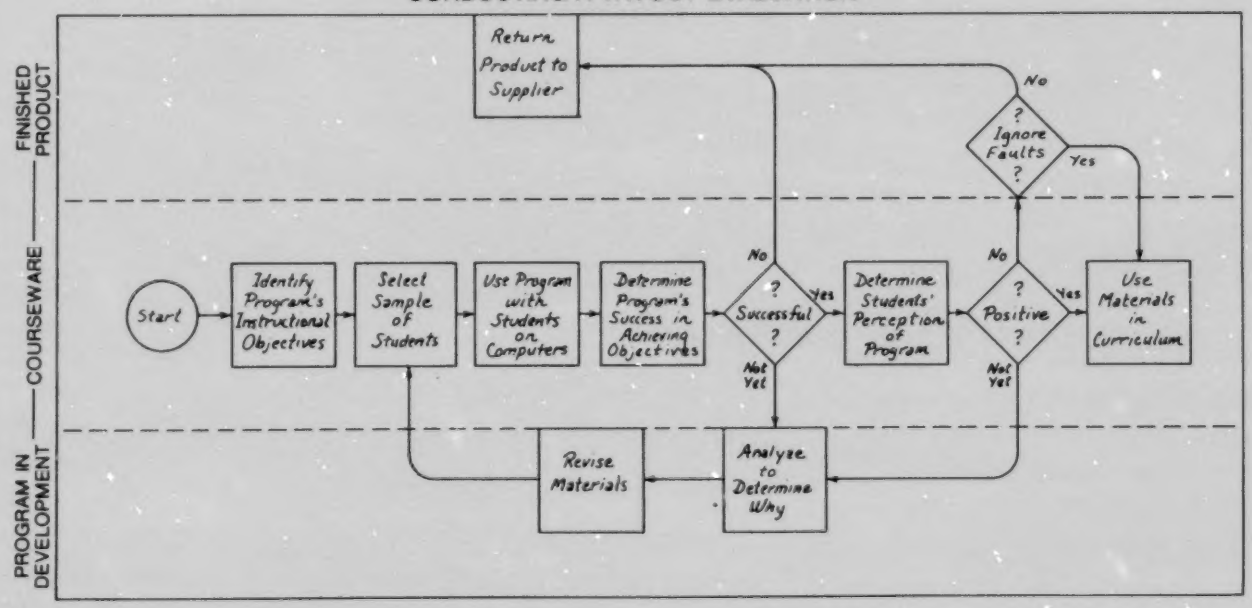
CONDUCTING A TRYOUT EVALUATION

The criteria utilized in a courseware review process such as that just described should be thought of as indicators, not proof, of courseware quality. "The proof of the pudding," you will remember, "is in the eating." An evaluation procedure consisting solely of a review of the type we have outlined thus far is incomplete. No matter how well-designed a computer program intended for classroom use may be judged to be, it still may not "work" in the classroom. Real evidence for the effectiveness of CAI and CMI is to be obtained only one way--through actual tryout of the materials with the intended users in the instructional setting. Figure 3 illustrates an empirical approach to courseware evaluation.

One of the powers of developing one's own materials lies in the ability to perform tryout evaluations of each component while the materials are being developed. When a student-based testing and revision process is employed, you are able to identify flaws and failures of product performance at early stages and make revisions to improve the product with less cost. During a courseware development process employing such formative evaluation principles, student tryouts can be performed with any program component in a sequence such as this: 1) observing one student during a session with the materials and making revisions accordingly; 2) trying out "semi-polished" materials with a small group of students using a semi-systematic questioning process based on criteria deemed important (selected from Tables A through F), and revising the materials based on observations and feedback; and 3) trying the "polished" materials in a normal classroom setting in a systematic fashion, making revisions if warranted.

Since such a "formative approach" is applicable only when revision is possible, student-based tryout of commercial courseware must be conducted with a finished product. The trial group of students selected to use the program should be sufficient in number to provide a degree of reliabity appropriate to the particular decisions concerning the

FIGURE 3
CONDUCTING A TRYOUT EVALUATION



courseware and its use. All should be members of the target population for which the courseware would be obtained.

As you can see from Figure 3, an empirical tryout of CAI courseware can have two aspects: 1) gathering feedback from students who use the materials, and 2) ascertaining instructional effectiveness of the program. While the former oftentimes offers valuable information to the evaluator, it is the latter that is the essence of the empirical evaluation of CAI materials. The real question is this: When the students use the materials, do they achieve the program objectives? After all, promoting student learning is the reason for obtaining CAI courseware in the first place.

TABLE G

Other Evaluation Factors

* Does documentation identify prerequisites for program use?
* Does the documentation offer evidence of systematic development, field testing, and program effectiveness?
* Does documentation provide suggested usage strategies?
* Does documentation provide a sampling from typical program run(s)?
* Is there a teacher's guide to the materials?
* Is the guide clearly written and easy to use?
* Does the guide include relevant adjunct instruction or information (e.g. student exercises, handouts)?
* Does the guide include correlation with standard texts?
* Does the guide suggest references/ topics/activities for follow-up?
* Is the program itself easy to use?
* Are operating instructions clear and complete?
* Does the program run on the computer as promised?
* Does the program appropriately handle erroneous entries?
* Is the teacher free from the need to intervene or assist?
* Is there support for the teacher from the vendor if a problem arises with the program?
* Are user support materials (e.g. student workbooks) appropriate and adequate?
* Is the program stored on high-quality disk or cassette?
* Can the program be reviewed prior to purchase?
* Can the program be duplicated if desired?

You can use the criteria in Tables A through F as guides to constructing questions to ask the students. You can determine how well they liked any lesson(s) and what suggestions they have for improvement. You can find out how valuable the materials are from their perspective. The questions asked, and the manner in which you ask them (orally during their tryout, on paper subsequent to the tryout, and so forth), will vary depending on the nature of the program and the number of students involved.

Of course, the most important question about instructional courseware that the trial group of students will be answering for you is this: Does the program "work"? That is, does it teach? The answer will come not from their opinions and suggestions, but rather from the learning that they evidence when using the program. You can ascertain it from the fraction of students attempting the computer lesson(s) who actually do, through their interaction with the courseware, learn what the materials are supposed to teach.

As with CAI materials, a CMI program also should be tried with representatives of the target user group to get their feedback and to ascertain whether or not the program is effective in doing the tasks which it will be looked to to perform in its intended setting.

Only after "proving its mettle" by successfully doing what it is supposed to do in classroom tryouts with a sample of intended users can courseware be said to have "passed" a courseware evaluation.

SUMMARY

We have presented a courseware evaluation model that can be applied during a design and development process or used when making a decision whether or not to purchase a program produced by someone else. Generally you will not want or need to utilize all of the criteria presented in the tables. You will want to select and choose those criteria that are applicable to a specific piece of courseware and a particular setting. Our intent has been to provide you with a fairly comprehensive framework from which you can select the questions you need to ask of your materials. Care must be taken not to skip over important criteria simply because a program "looks good" on the surface. If a computer program is going to be used in a serious manner with a fairly large number of students, it merits close professional inspection. The students deserve no less.

AUTOMATED HOLISTIC SCORING OF ESSAYS

Vernon Miles and Sue Legg, Ph.D.

University of Florida

ABSTRACT

This paper describes the automated holistic scoring process for the College-Level Academic Skills Test (CLAST) in the state of Florida. Approximately 15,000 papers are read twice and split scores are resolved for each of the three administrations of the examination. The system uses NCS 7001 optical scanners and Apple II + micro-computers. The rationale, systems design, technical problems and evaluation are discussed.

The automated system is an improvement over current computer batch processing. Batch processing reduces the clerical errors associated with manual scoring. However, a second reading session must be scheduled to resolve discrepancies in scoring. Automated on-site scoring provides information that can be used during the initial reading to control reliability and to assist referees in resolving discrepant scores. Technical issues addressed in on-site scanning include the communication between the scanners and micro-computers, increasing the speed of processing and "garbage collection" in micro-computers.

INTRODUCTION

In 1982 the Office of Instructional Resources at the University of Florida accepted the task of managing the holistic scoring of essays written for the College-Level Academic Skills Test (CLAST). Three times each year approximately 15,000 papers are read twice by eighty to 120 readers during a two day period. Scores on these essays are assigned on a four-point scale. The two scores are added and become part of the admission requirement to the upper division of the State University System. The use of scores as admission criteria intensifies the need for an efficient, reliable scoring procedure. This paper describes the automated, holistic scoring system that was designed for the CLAST. The system uses NCS 7001 optical scanners and Apple II + micro-computers. The rationale, systems design, technical problems, and evaluation are explained.

RATIONALE FOR THE AUTOMATED SYSTEM

Holistic scoring is generally conducted as a manual process or manually with computer batch processing. Essential for the success of the process are the secrecy of the scores, the identification of disagreements on scores between readers (called splits), and the reliability of the scoring. Readers must adhere to the writing standards that represent each score level, or the consistency in scoring is lost and passing or failing becomes an arbitrary decision.

The manual method has several serious limitations that precluded its use for the CLAST. Among these problems are: 1) It is labor intensive with an average of one staff person to process each 1,000 forms 2) Manual addition of reader scores is error prone 3) Identification of splits is error prone 4) Feedback on reader reliability is qualitative rather than quantitative. No hard data can be kept during the reading session on splits attributed to individual readers or on splits as a percentage of essays read.

Existing batch computer processing eliminates much of the error in scoring but does not give on-site feedback on reader reliability. In addition, splits must be resolved by a third reading which has to be scheduled at a later date. For these reasons, the decision was made to

test the feasibility of an on-site scanning process. Two sites were designated, one for automated processing and one for manual plus batch processing.

Two major factors made the automation of the holistic scoring process a possibility: 1) the availability of reliable, low cost, portable micro-computers, and 2) the introduction of a low cost portable optical scanner compatible with existing optical mark sense forms used on larger permanent site scanners. The Apple II + micro-computer and the National Computer Systems 7001 table top scanner were chosen as the combination to automate the holistic scoring process for the CLAST.

SYSTEM DESIGN

The automated holistic scoring system for the CLAST has three components. The first part is a machine readable optical mark sense form attached to the back of each essay booklet (Fig. 1). This form is used to record the essay topic selected by the student at test time, the reader identification numbers, and the scores assigned during the reading session. The second part of the system is the NCS 7001 optical mark scanner. The essay score sheets are fed through the scanners at the reading site. The scanners check the form for bubbling errors and omission of required data. Valid data from the forms are passed on to the third part of the system, the Apple II + micro-computer. The Apple computer converts a letter score code to a numeric score of one through four for each of the two readers. A difference of more than one between the two scores is immediately identified as a split, and the score sheet and essay booklet are sent to a referee reader to resolve the split. Successfully scanned forms are recorded in the micro-computer to provide on demand reader statistics and to transmit to a larger host computer where the data is merged with other test data before final processing.

This system has several features and advantages which make it desirable over other methods. First, the score sheet only identifies the paper by an answer sheet number. This precludes the reader from being biased in any way by the student's school, name, or any other biographical data which is often unsuccessfully masked out on score sheets with other systems. Second, readers are assigned unique four letter code words (Fig. 2), which represent the score numerals one through four. Encoding the scores in this manner ensures that no reader will know what another reader has scored for the paper. As Figure 2 shows, almost any four letter word can be used as a reader code. Third, reader code errors are caught by the system in time to correct them during the reading session. Fourth, and most important, reader splits are identified during the reading and not after, eliminating a second reading session. It is also possible to monitor the percentages of splits at any time so that consistent reader reliability can be maintained by retraining as necessary.

The system configuration consists of two Apple II + micro-computers with two NCS 7001 scanners attached to each Apple. A fifth scanner is held in reserve.

Each Apple computer runs a series of five different programs that are described below:

Code Access -- Used well before scanning begins to generate the four letter table codes used by the readers. It allows manual or automatic generation of codes.

Reader Access -- Allows the user to initially assign readers to specific reader/table numbers. No-shows and table changes can be quickly updated using this program at the reading site. This program can also be used at any time after scanning has begun to determine reader counts for individual readers.

CLAST Scan -- Performs the majority of the work at the reading site. It receives data from the scanner, checks the data for various errors, writes valid records to disk storage, and accumulates totals on reader scores.

Reader Stats -- Provides the administrators of the holistic scoring

session with reader reliability statistics. Score counts and percentages are given for each reader and for the session as a whole.

Trans CLAST Data -- Transmits data stored on diskette (by the CLAST scan program) to the host computer at the University of Florida campus. This program interfaces with a TSO CLAST program at the host computer.

As of this writing the system has been used three times to process 40,000 forms. Each scanner when manned by two operators can comfortably process 500 forms per hour for a total system capacity of 2,000 forms per hour. Additional personnel are assigned to separate the essay score sheets from the booklets before scanning and to sort the essay score sheets and booklets after scanning. The operation requires fourteen people to work twelve hours to scan, process errors, and sort forms and booklets. However, it should be noted that the time required to process the forms is more a function of the return rate of forms from the readers than a function of the system capacity.

TECHNICAL PROBLEMS

There were many technical problems to overcome in implementing the project. The first problem to be solved was the communication between the Apple and the scanners. Fortunately, the Registrar's office at the University of Florida had already done considerable research and development while implementing a new registration system. Communications protocols and basic communications routines were written. Using the Registrar's programs as a starting point, modifications were made to work with our equipment.

Once a one scanner, one Apple system was working, it was decided to try for a two scanner, one Apple system to reduce cost and reduce the component count of the system. The communications protocol of the NCS 7001 scanner is a X-ON, X-OFF protocol that turns the scanner OFF while it is processing the data from the current form. The scanner also allows a record to be retransmitted to the Apple on demand. By using a routine in the Apple CLAST scan program to peek at the data input buffer of the Apple communications card, it is possible for one Apple to scan the input buffers of several communications cards to determine if 1) a record has been transmitted by a scanner 2) a scanner is currently transmitting 3) a scanner has completed transmitting or 4) no record has been received. This routine means that one Apple can scan several scanner communications cards in a multiplexing mode while looking for a scanner that has completed transmission of a record. At this point, the multiplexing routine stops, and the scanner is commanded to resend the record. The CLAST scan program then captures the record, processes it, and instructs the scanner to release the current form and accept the next. The scanner multiplexing loop is then re-entered and the Apple "looks" at the next scanner in sequence. Figure 3 is a flow chart which better illustrates this technique.

The initial run of the system revealed one irritating problem which, while not fatal, was considered unacceptable to performance. Most micro-computers (the Apple included) store data into available memory space. As new data is received, rather than replacing the existing data, the micro-computer simply moves to another area of "free" memory and saves the new data. Eventually, the "free" memory is filled, and the computer appears to stop working while it goes into what is appropriately called "garbage collection." The frequency that this happens is a function of the micro-computer memory size and the size of the program running in the computer. In the case of our system, "garbage collection" occurred approximately every sixty forms and lasted almost two minutes. This delay resulted in a loss of almost 40 percent of system capacity. While there was enough spare capacity to handle the loss, it was considered unacceptable for future needs.

The solution to the problem was a compiler program called Expediter II by on-line systems, Coarse Gold, California. After only minor changes to our basic program, Expediter II converted it to machine code that reserved memory areas for specific variables. New data could be swapped out with old data so that "garbage collection" was unnecessary. As an extra bonus, the compiled version of the program runs approximately four times faster than the standard basic version which allows faster processing of the forms.

While designed specifically for Apple II + micro-computer and the NCS 7001 scanner, the concepts presented in this paper are not limited to this equipment. Recent advances in micro-computer equipment and in lower cost scanners make duplication of this system even more affordable.

EVALUATION

The purpose of the evaluation was to establish the feasibility of the system. Several concerns were to be monitored: the adaptability of a mechanical process to the procedures used for reading the essays, the reliability of the equipment, the usability of the data, and the cost effectiveness. This adaptability issue was perhaps the most fundamental. Automated scoring imposes constraints on the flow of papers in order to keep the scanners active and to avoid a backlog of papers to be processed. However, control of holistic scoring must be lodged with the referees who plan the breaks in reading and schedule retraining sessions. The computer scoring must adapt to the needs of the readers rather than vice versa.

The trial run of the system led to several conclusions about the feasibility of the system. The regulation of the flow of papers to and from the readers was complicated by the automated procedures. Previously all first readings were completed before second readings began. This procedure had to be changed, because scanning could not occur before papers had been read twice. The advantage obtained by monitoring the readings outweighed any problems caused by restructuring the reading process. Equipment reliability had to be guaranteed by having backup machinery available. The usefulness of the data to monitor scoring reliability was immediately apparent. The continuous flow of information about the quality of the scoring enabled the referees to assist individual readers and to retrain the group as necessary. Cost-effectiveness could only be established if the equipment had multiple uses in addition to the holistic scoring. At our institution, the scanners were already being used for registration, and the micro-computers were also available. Clearly, the automated system was superior to alternative methods as long as the technical staff and equipment were readily available.

A second-level evaluation is planned which will compare the reader reliability data from readings located in three different sites. One of these sites will use the automated system and the other two will use the computer batch system. The objectives of this evaluation will be to assess the effectiveness of the immediate feedback on improving reader reliability.

Interfacing Your Microcomputer to the Real World

Robert S. Tannenbaum

Hunter College of the City University of New York

David C. Tannenbaum

Rensselaer Polytechnic Institute, Troy, New York

Introduction

Personal computers are not just for instruction, computation, and playing games. You can control a simple circuit or a complex experiment. You can turn on or off any number of audio-visual devices, such as a tape recorder, slide projector, or movie projector, in any sequence you require. You can control any or all of the lights in your classroom or laboratory at any time of the day or night whether you are there or not. You can detect an event and sound an alarm or turn on an automatic response system. If what you want to sense or start or stop contains a switch, you can use a micro- computer to detect its status or control it by having the computer turn it on and off. It takes some very inexpensive interfacing and just a bit of experimenting. The only things which are required are that your computer have a latched parallel input/output port that can transmit and receive bits of data at TTL levels (about +5 V) and the input and output circuits which are detailed in this paper. The circuits may be built singly from very inexpensive parts. If you need more than one or two circuits, you may combine and multiplex them on a printed circuit board very efficiently. The boards and the parts are available as bareboards, as kits, or assembled at quite reasonable prices. For large input and process-control applications where you require more than one input and/or output board, a mother board, which will accomodate up to eight input and eight output boards, can be made. The mother board is also available in bareboard, kit, or assembled and tested form.

Parallel Input and Output

In order to use the sensing (input) and control (output) circuits we have developed, it is necessary for your computer to have a latched parallel input/output port and for you to be able to access the individual bits of data processed via that port. All common personal computers provide such a port and such access either as a standard or an optional capability. We are including a brief review of parallel I/O here to provide you with the necessary background for implementing the sensing and control circuits.

Digital information which is to be transferred into or out of a computer is respresented in binary form, that is as a string of 1s and 0s. Each 1 or 0 is called a bit (for Binary digIT.) The string of bits may represent any of several things:

1. It may represent a number (coded in some numerical code such as binary, binary-coded decimal, etc.).

2. The string of bits may also represent a character (a letter, numeral, or special character) again coded in some way (such as ASCII, EBCDIC, etc.)

3. Or the string of bits may carry some other information which controls or indicates the status of several switches connected to the parallel port.

What makes these parallel I/O is that for each bit there is a separate data line (wire) which carries that bit at the same time as (and in parallel with) each of the other bits in the data.

When numerical or character information is to be transferred to and from the parallel port, most higher level languages handle all of the coding automatically. In BASIC, for example:

```
10 PRINT "B"
```

will result in the transfer to the parallel port (if it is the primary output device) of the bit string 01000010. This is the ASCII code for the upper case letter B. Likewise,

```
20 INPUT N
```

will result in the numerical value of whatever is entered to be stored in the memory location associated with the variable named N. If you entered the numeral 9, then the bit string 00001001 would be stored.

It is possible for you to use this automatic encoding to create and transfer the bit strings necessary for sensing and control. However, it is very awkward to do it that way. It is far more convenient to use the PEEK and POKE facilities provided in most higher level languages. Only if these are unavailable should you have to resort to the standard PRINT and INPUT instructions.

PEEK and POKE are primative operators. They are the equivalent of the machine language command to MOVe data from one location to another.

> **PEEK** enables you to move data from any location in the computer to the location associated with a variable named by you in the higher level language.

For example:

```
10 D = PEEK(1234)
```

results in whatever is currently stored at location 1234 being MOVed to (i.e., replicated at) the memory location associated with the variable D. Now you can make use of the data via the higher level language merely by using the variable D. PEEK is a move from an unnamed computer location to one named by you in the higher level language program.

The reverse is true for POKE.

> **POKE** enables you to move data from a memory location associated with a variable named by you in the higher level language to any location in the computer.

For example:

```
20 POKE(5678, V)
```

results in the transfer of whatever is currently stored at the memory location associated with variable V to location 5678.

POKE is potentially a very dangerous instruction. Since you can POKE any location in the computer, it is very easy to destroy essential system information by accident and cause a system crash. You should take extreme care when using the POKE command. Be absolutely certain that you have the address specified correctly and that altering its contents will not affect the operating system.

Of course, different higher level languages and even different dialects of the same language implement PEEK and POKE somewhat differently. You will have to study your system language manual(s) to determine how these instructions are implemented on your computer. But the effect in all systems is the same:

> transfer of data between a named and an unnamed computer location.

In the unlikely event that your system does not already have a PEEK and a POKE facility available, if it gives you the ability to write assembly language functions and/or subroutines then you can create them for yourself without too much trouble.

In order to use the PEEK and POKE routines as we want to (to sense the status of and control individual external binary switches), it is necessary to specify the meaning and purpose of each bit in the byte or word being transferred. This is called "bit mapping."

If you needed to turn on devices 4 and 5 , you would have to cause the bit string 00011000 to be POKEd to the location of the parallel output port.

If your higher level language permits binary variables then this is no problem. However, most higher level languages (unfortunately) do not allow binary variables. Therefore, you will need to do some simple binary-to-decimal conversions.

The number 00011000 in binary converts to 24 in decimal. So you would need to POKE decimal 24 to the parallel output port to turn on devices 4 and 5 and leave all the others turned off. If your parallel output port were located at 12346, then you might do this with the command:

```
10 POKE(12346,24)
```

The reverse is true for sensing the status of switches connected to a parallel input port. Assume that the address of the port is 12345 and that a 1 indicates a switch is on and a 0 indicates it is off. And assume further that the following instruction was executed:

```
20 S = PEEK(12345).
```

If the resulting value of S were 35 decimal, then a decimal-to-binary conversion would be necessary to determine which switches were on and which were off at the time of the PEEK.

The number 35 decimal converts to 00100011 in binary. This indicates that at the time of the PEEK only switches 1, 2, and 6 were on and switches 3, 4, 5, 7, and 8 were off.

It is a relatively simple exercise to write binary-to-decimal and decimal-to-binary conversion routines in a higher level language. Many introductory programming texts give examples or algorithms.

Sensing the Status of a Switch

You can use the PEEK command to determine the status of the parallel input port of your computer. Now it is time to find out how to connect that port to the real world. You need some sort of single-pole, single-throw switch which changes position (status) in response to the real world event you want to monitor. For example, if you

want to check for an open window in a primate cage, you can mount a wiper switch on the frame so that it is closed (makes contact) when the window is closed and is open when the window is open. Or, if you want to sense when the temperature in your laboratory greenhouse goes above a certain level, you can use a heat-controlled relay (a thermostat).

The switch has to be connected in a circuit to one of the bits of the parallel input port. When the switch is in one position, the port receives a 1 and when the switch is in the other position, the port receives a 0. The simplest circuit to do this is a series circuit containing the parallel input port, the switch, and a current source (such as a battery). This circuit is the basis for the complete system shown in Figure 2.

It would be nice if the simple circuit worked well in the real world. Unfortunately, it works best only in theory and on paper. Two factors require us to modify this simple circuit to make it work reliably. One is differing voltage requirements at different points in the circuit. The other is stray electrical noise in the circuit. It will remain a series circuit in effect, but we will introduce some additional elements to compensate for these factors.

Limiting Current and Voltage

The first factor is that different parts of the system may require different currents and voltages. The parallel input port requires "TTL voltages." TTL stands for Transistor-Transistor-Logic. It is the form of circuitry used in standard digital logic chips such as the 7400 series. These devices need +5 V. and ground with no more than 10-25 mA of DC current. But this may not be enough for the rest of the circuit if you have long wires (your switch is a long way from your computer) or if your switch is in a dirty environment. To avoid significant losses over longer wires, you need to use a higher voltage. And to keep your switch contacts clean, a higher voltage and current will help by arcing each time the switch is opened or closed. The tiny arc will burn off any dust or dirt.

So, we need a circuit with TTL levels at the parallel port and higher levels along the feedlines and at the switch. To do this we add current and voltage limiting resistors to our circuit. A simple circuit consists of two resistors. One resistor goes in series between the source (our switch) and the destination (our input port). The other resistor goes to ground. Part of the current flows through this second resistor leaving only as much as we need for our port. R1 and R2 in Figure 1 constitute the limiting ciruit.

In order to calculate the resistor values required in the limiting circuit, you must decide the voltage and current needed for the switch and feed lines of the circuit. These depend on the length of the lines and the environment of the switch. We have used 24 V. and 100 mA. successfully. You may need to do a bit of experimenting to determine the best levels for your system. After you have chosen the values, you will use them in the resistor calculations. We will do this here with symbols and then we will substitute some real numbers in an example. (Refer to the schematic portion of Figure 1.) We will use the standard direct current equations and symbols for:

E -- Voltage measured in Volts
I -- Current measured in Amps
R -- Resistance measured in Ohms
P -- Power measured in Watts

The subscript t indicates that the value is for the total circuit
A numerical subscript indicates that the value is for only one component in the circuit
The subscript p indicates that the value is for the input port
The subscript p1 indicates that the value is for the input port leg of the circuit

The equations you need are:

1. $E = I \times R$; $I = E/R$; $R = E/I$ (Ohm's law)

2. $P = I \times E$

3. $R_t = R_1 + R_2 + \ldots$
for resistors in series

4. $1/R_t = 1/R_1 + 1/R_2 + \ldots$
for resistors in parallel

You may need to rearrange the formulas algeoraically to solve for one of the variables.

Assume that the total voltage you want at your switch is E_t and the total current is I_t. E_p and I_p are determined by the requirements of the port you are driving. Since R_1 and the port are in series on one leg of the circuit, the current through each of them is the same and the sum of their voltages must equal the total circuit voltage. These relationships may be expressed as

$$I_1 = I_p$$
$$E_1 = E_t - E_p \text{ (because } E_t = E_1 + E_p)$$

Now you can solve for R_1.

$$R_1 = E_1 / I_1 = (E_t - E_p) / I_p \text{ (using equation 1.)}$$

and

$$P_1 = I_1 \times E_1 = I_p \times (E_t - E_p) \text{ (using equation 2.)}$$

R_2 is in parallel with the combination of R_1 and the port. Therefore the voltage across it is the total voltage and the current through it is the remainder after I_p has been subtracted from the total current. These relationships may be expressed as

$$E_2 = E_t$$
$$I_2 = I_t - I_p \text{ (Because } I_t = I_p + I_2)$$

Now you can solve for R_2.

$$R_2 = E_2 / I_2 = E_t / (I_t - I_p) \quad \text{(using equation 1.)}$$

and

$$P_2 = I_2 \times E_2 = (I_t - I_p) \times E_t \quad \text{(using equation 2.)}$$

To prove that these results are correct, you can now substitute the results for R_1 and R_2 in the formulas and show that the total resistance for the circuit is equal to E_t divided by I_t. That is,

$$R_t = E_t / I_t$$

This may be shown as follows:

1. The resistance in the input port leg of the circuit is the sum of R_p and R_1 because they are in series (equation 3.)

$$R_{p1} = R_p + R_1$$

2. The inverse of the total resistance is equal to the sum of the inverse of R_2 plus the inverse of R_{p1} because they are in parallel (equation 4.)

$$1/R_t = 1/R_2 + 1/(R_p + R_1)$$

3. Substitute the calculated values for R_2, R_1, and R_p in the line above.

$$1/R_t = 1/E_t/(I_t - I_p) + 1/[(E_t - E_p)/I_p + E_p/I_p]$$

4. Simplify

$$1/R_t = (I_t - I_p)/E_t + 1/[(E_t - E_p + E_p)/I_p]$$

$$1/R_t = (I_t - I_p)/E_t + I_p/E_t$$

$$1/R_t = (I_t - I_p + I_p)/E_t$$

$$1/R_t = I_t/E_t$$

5. Invert the equation

$$R_t = E_t/I_t \quad \text{Q.E.D.}$$

You should familiarize yourself with the computations for determining the values of the resistors in the limiting circuit. Now we will use some real numbers and solve a sample circuit to get the values for the resistors. Assume that your port requires 5 V. and 10 mA. (10 mA is equal to .01 Amps.) Also assume that you want the total voltage to be 24 V. and the total current to be 100 mA. (or .1 Amps).

$$R_1 = (E_t - E_p)/I_p = (24 - 5)/.01 = 19/.01 = 1900 \text{ Ohms}$$

$$P_1 = I_p \times (E_t - E_p) = .01 \times (24 - 5) = .19 \text{ Watts}$$

$$R_2 = E_t/(I_t - I_p) = 24/(.1 - .01) = 24/.09 = 266.667 \text{ Ohms}$$

$$P_2 = (I_t - I_p) \times E_t = (.1 - .01) \times 24 = 2.16 \text{ Watts}$$

Since there is some flexibility in the voltage and current requirements of the port, you can round these results to easily obtainable standard values. Be sure to round the power ratings (Watts) UP so that the resistors will be able to dissipate the heat without burning out. Our sample values could be rounded to

R_1: 1.8 k Ohms or 2.0 k Ohms at 1/4 or 1/2 Watt

R_2: 270 Ohms at 4 or 5 Watts

Opto-isolation

The second factor we have to compensate for is stray electrical noise which may be introduced along the feed lines. Any long conductor acts as an antenna and small currents may be induced in it by stray electro-magnetic fields. Such fields are present everywhere, but they are strongest and most common in areas where a lot of electrical activity is taking place (for example, where motors are starting and stopping). A centrifuge, a refrigerator, a video tape recorder, and even your personal computer all cause fields. Some of these may be strong enough to induce enough current in your feed lines to cause false positives at your input port. You can try to shield the feedlines, but that is very expensive and not always very effective. What we need is another addition to our circuit to separate the feedline current from the input port current. We use an opto-isolator.

An opto-isolator (sometimes called an opto-coupler or a photo-coupler) consists of an infrared light emitting diode (IRED LED) and a light sensing device sealed together in a light-tight package. When the LED is forward biased, it gives off light. The light sensing device will not conduct in darkness. It must be lighted or it behaves like an open switch. When the circuit on the light emitting side of the opto-isolator is completed, the LED gives off light. The light sensing device "sees" this light and begins to conduct. This completes the circuit on the other side of the opto-isolator. Electrically the two circuits are isolated. They are coupled by the light which is emitted from one side and "seen" by the other. Small stray currents induced on the LED side of the opto-isolator will not produce sufficient light from the LED to activate the light sensing device. In this way the second circuit is isolated from electrical noise induced in the first circuit.

We insert the opto-isolator in our system between the limiting circuit and the input port. This is shown in Figure 2. There are several types of opto-isolators available. Here you should use one with either a photo transistor output or a photo Darlington output (for example, Radio Shack 276-1628 or G.E. 4N26 or 6N26). Connect the limiting circuit and ground to the anode and cathode of the LED. Connect the input port and ground to the collector and emitter of the output transistor. Leave the base of the output transistor unconnected. See Figure 2.

You will have to recompute the values for the resistors in the voltage divider using the requirements for the LED. Most opto-isolator LEDs need about 20 mA at no more than 1.7 V. With these values and E_t = 24 V, and I_t = 100 mA., after using the equations above and rounding to standard values, we get:

R_1: 1200 Ohms at 1/2 Watt

R_2: 300 Ohms at 2 Watts

Try it for yourself to be sure you get the same answers.

You should note that the opto-isolator acts as an inverter, too, because of the action of the output transistor. So now when the switch is closed the computer receives a 0 and when it is open the computer receives a 1.

Embellishments

There are three additional elements which you can add to your circuit. These are:

1. a fuse for protection,
2. an LED for visual confirmation of the status of the switch, and
3. a debounce circuit for clean, fast recognition of a switch change of status.

Fusing

The fuse goes in series with the switch before the limiting circuit. The total current (I_t) flows through the fuse. In our example this is 100 mA. You should use a fast-blow fuse which will handle the total current, but will blow if that current approximately doubles. A standard value fuse which is readily available is 1/4 Amp (250 mA.) That is close enough for our needs. Remember that the fuse must be rated to handle the total voltage (E_t), also. In our example that is 24 V. Figure 2 shows the circuit with the fuse added.

LEDs

You can install an LED before or after the opto-isolator to indicate the status of the switch. If you install it before the opto-isolator in series with R_1, it will respond directly to the switch. You can also install it in parallel with the opto- isolator. In each case you have to recalculate the values for R_1 and R_2 including the voltage and current requirements for your new LED. After you recalculate the values, you can experiment with slightly different values to increase or decrease the brightness of the LED to suit your taste.

What happens if the opto-isolator malfunctions and does not retransmit the signal from the switch circuit?

1. If you install the LED before the opto-isolator, it indicates the status of the switch, but it does not indicate what is being transmitted to the input port so it does not signal the malfunction.

2. If you put an LED after the opto-isolator, it will indicate the status of the opto-isolator. It will go on only when the opto-isolator is on so it will signal a malfunction.

3. You can install both LEDs in your circuit as a double check, as shown in Figure 2. When they both light together, you know all is functioning properly. However, if only the first one lights, you know the optoisolator is malfunctioning.

Debouncing

All mechanical switches have a physical characteristic called "bounce." Imagine you are landing on a trampoline. You don't just hit the surface and stick there. You bounce up and down several times before settling firmly to rest. The same is true of a switch. It doesn't just close and make electrical contact in one clean action. Electrical contact may be made and broken several times before the switch "settles" into its final on state. This making and breaking of the circuit as the switch is closed is called "bounce."

The length of time required for the switch to settle may be long compared to the time required by your computer to PEEK at the circuit. This means that if you PEEK at your circuit while the switch is bouncing, you may see it either on or off and you won't know for sure which is correct. There are two ways to correct for this problem -- using hardware or using software. It is possible to insert a debounce circuit between the switch and the computer. This usually consists of either 1) an OR gate plus a one-shot, or 2) a flip-flop. You would need one such circuit for each switch you want to debounce or a way to route each switch through the same debounce circuit. We decided this would be awkward and expensive and so we did not add <u>hardware</u> debounce components to our circuits.

The same debouncing can be accomplished via software. It is simple, but it requires at least twice as much time to do each test of a switch. In a software debounce, the program PEEKs at the circuit twice and compares the two results. If the results are equal, the program assumes the switch has settled and the result of the PEEK is correct. If the results are not equal, the program assumes the switch is bouncing and keeps PEEKing repeatedly until it gets two successive results which are equal. Clearly, this must take at least twice as long as a hardware debounce circuit. But in most applications the time delay is so slight compared to other actions the program must take that the slowing makes no practical difference. So we recommend a software debounce solution.

Controlling Outside Systems

Just as you can use the PEEK command to determine the status of the parallel input port, you can use the POKE command to send bits to the parallel output port. These bits may be used to control systems outside of the computer by turning them on or off. For example, if your input system senses that it is getting dark outside, you could have the output system turn on your laboratory

greenhouse gro-lights. Or, if your computer PEEKs at your solar hot water system and finds some combination of conditions to be true (e.g., storage tank temperature too low and solar panel temperature high enough), it can POKE a bit to the output port which will cause the circulating pump to be turned on.

The device outside of your computer must be connected in a circuit to one of the bits of your parallel output port. The bit serves as the switch to turn the device on and off. If the bit is set to one by the computer, it turns the device on. If it is set to zero, it turns the device off. As with the sensing circuits, the simplest circuit to achieve this would be a series circuit.

The only devices which can be controlled directly by a such a simple series circuit must have voltage and current requirements at TTL levels because that is all that the output port can control. But most of the devices you might want to control have quite different electrical requirements. In particular, most of the devices need AC current at 120 volts. So what is needed is a way that the bit at the output port can open and close a switch in an AC circuit.

There is a simple electronic circuit which can be used to achieve just this function. It consists of an optocoupler and a triac. A triac permits a low-voltage AC signal to switch a much higher voltage AC circuit. The optocoupler serves the dual functions of

(1) coupling the low-voltage (TTL level) DC signal that we have at our output port to the low-voltage AC signal and

(2) isolating the port from any stray voltages on the output lines.

(You can see now why the same device is known both as an opto-coupler and an opto-isolator.)

As the output DC signal from the port goes on and off, this causes the low-voltage AC signal to go on and off because they are coupled. This, in turn, controls the higher-voltage AC signal via the triac. Figure 3 shows the control circuit with the optocoupler and triac in place.

The resistors in Figure 3 are required to limit the current and voltage to the triac driver portion of the opto-coupler. They also help increase the reliability of the circuit by eliminating false triggering of the triac. Their values depend on the particular triac you use. The triac should come with suggested values. For example, Radio Shack part number 276-1000 (a 4 amp triac) recommends 180 ohms and 47 ohms.

DANGER

WARNING: On the output circuits, you are dealing with 120 volts AC. This can cause severe electrical shock. Do not touch or service the circuits unless they are disconected from the AC source.

Embellishments

The simple circuit will permit you to control any 120 volt AC device from one bit of your parallel output port. All you have to do is POKE that bit to 1 to turn the device on and POKE it to 0 to turn the device off. But you may again wish to add some embellishments to your circuit.

It is always prudent to insert a fuse in a circuit. Remember that now you are fusing a 120 volt AC circuit so the fuse you choose must be rated for that voltage. The amperage of your fuse depends on the device you are controling. It should be high enough to permit the device to function normally, yet low enough to burn out before any harm can come to the device in an emergancy. Most devices are clearly labeled as to their current reqirements or come with specifications. (Be sure they do not exceed the amperage rating of the triac.)

You may also wish to add indicator lamps to your circuit. In the DC portion of the circuit you can again use LEDs with an appropriate resistor. If you want an indicator lamp in the AC portion of the circuit, you should use a miniature neon lamp such as an NE44. The neon lamp is designed to operate on 120 volt AC current.

Snubber Circuit

If the AC load you are controlling is inductive, it may generate voltage spikes across the triac which can cause false turn-on of the circuit. To avoid this, a snubber circuit may be added. The snubber consists of a resistor and a capacitor in series placed in parallel with the load. Figure 3 shows the complete output control circuit with the snubber circuit added to it.

Latching

Even with the embellishments, the circuit in Figure 3 may not meet your needs. Some output ports are not latched. That is, the 1 or the 0 that you POKE out to the port does not remain there after the POKE instruction is completed. The 1s and 0s are present at the port for a fraction of a second during the POKE. Then the bits go back to an idle state which is neither a 1 nor a 0. Even if your output port is latched and the bits remain as 1s or 0s until changed by another POKE, you may wish to use the port to control several devices. This calls for a circuit which will remember the bit(s) for one device while the port is being used to change the bit(s) for other device(s).

What is required is something called a latch. It is a device which will stay in either the 1 or the 0 state until POKEd again. The latch is installed between the output port and the opto-coupler. The state of the latch is set and changed by changes in the output port. When the latch is enabled and the output port bit goes to a 1, the latch goes to 1 and stays there until it is enabled again and the ouput port goes to a 0. And vice-versa, when the latch is enabled and the ouput port goes to 0, the latch stays at 0 until it is enabled once more and

the output port goes to 1. In this way, via the coupling of the TTL level output of the latch to the triac control signal, the higher-voltage AC circuit stays on or off until you change it with another POKE command to the output port.

There are numerous integrated circuits which may be used as latches. We are using a 74259 as our latch. We chose this particular device because it has eight addressable latches in one chip. This makes it possible for us to control more than one higher-voltage AC circuit from the same output port bit. We will discuss this in the section on multiplexing. To ensure that the 74259 is not overtaxed by driving the opto-isolator and the indicator LED and to invert the signal from the 74259, we use a 7404 (hex inverter) after the latch. Figure 4 shows the multiplexed circuit with the latch and the inverter added. If you are controlling only one or two devices, you may wish to use a latch that does not require addressing, such as a 7475 or a 7477.

Multiplexing

If you need to sense or control more than just one or two systems, you will quickly run out of bits on your parallel input/output port unless you multiplex your circuits. Multiplexing is a technique which permits you to use the same circuit to handle multiple systems. This is accomplished by employing addressable circuit elements. You use several of the port bits as address bits to control the flow of one data bit to or from your systems. If you use three address bits, you can control eight different circuits. With three more bits, you can control sixty-four different circuits.

Multiplexing Input

You could sense the status of eight different switches by connecting eight of the circuits to eight different bits on your parallel input port. Or you could use just four bits and a simple multiplexing circuit. We used a 74152 eight-input multiplexer chip. Figure 5 shows the circuit.

Now you are using both input and output bits to sense the status of the switches. You are using three output bits to send to the 74152 the address of the bit that you want to sense. And you are using one input bit to read the status. This circuit requires that you POKE the address to the output bits and then PEEK at the input bit to sense the status of the switch. You should note carefully that the status of the switch will be inverted again by the 74152. That is, now a 1 indicates the switch is closed and a 0 indicates it is open. This should provide no problem as long as you are aware of it when you are writing your software.

Remember our discussion of debouncing. You may wish to POKE the address bits and then PEEK twice and compare the results of the two PEEKs. If they are the same, you may assume that what you read is the correct status of the switch But if they are different, the switch is probably changing or bouncing and you should keep PEEKing until it settles.

If you want to connect more than eight circuits, you can use another 74152 chip. You may connect as many as eight of the circuits shown in Figure 5 as the inputs to this chip. You use three address bits from your output port to control this "mother" chip to select which of the eight "daughter" circuits you wish to read. You use another three bits to select which input line to the "daughter" circuit will be read. Thus, with a total of six output bits you can select among as many as 64 different input circuits.

Notice that the input signal is inverted once more by the "mother" 74152 chip. This means that now a 0 indicates the switch is closed and a 1 indicates that it is open. You must be very careful when writing your software to take into account the number of times the input signal has been inverted. If it has been inverted an odd number of times, it will be the inverse of what it started as. With no inversions or an even number of inversions, the logic state of the bit will be unchanged when it reaches the input port.

Multiplexing Output

Multiplexing output is very similar in theory to multiplexing input. In practice, however, there are complications imposed by the necessity of latching each output circuit. To multiplex as many as eight output circuits, you can use a 74259 chip. This is an eight-bit addressable latch.

Figure 4 shows the multiplexed output circuit using the 74259. In order to use this circuit, you must send the three address bits and the data bit to the chip . . . and then enable the chip. The practical complication is the slight time delay required before you enable the chip. If the address, data, and enable bits are sent simultaneously, there is no guarantee that the address and data bits will have settled to their correct logic states by the time the enable bit causes the chip to try to change one of its latches. This uncertain timing condition is known as racing and may lead to unpredictable results for the latches.

What is needed is for the enable bit to be sent just a tiny instant after the other four bits. If your output port has latched, separately POKEable bits this is no problem. All you have to do is POKE the first four bits before you POKE the enable bit. If you have two or more latched output ports, it is also very easy. Just use one port for the first four bits and a second port for the enable bit. Again, POKE the enable bit after the other four. The difficulty arises if you have just one output port and its bits are not latched or separately POKEable. Then you will need some circuitry to either remember the first four bits for a while or delay the action of the enable bit. This circuitry will probably depend on access to your computer's address bus (so you can decode it to know when you are addressing your output port) and your computer's write strobe. The circuitry will be machine-specific and a discussion of it is beyond the scope of this paper.

Further multiplexing the output circuits so you can control more than eight devices is analogous to what we did for the input circuits. In this case we use a 74138 as the "mother" chip. This chip is a 1-of-8 decoder/demultiplexer. It has eight output bits. These are used as the enable bits for the 74259 chips on as many as eight "daughter" circuits of the type shown in Figure 4.

In the multiplexed output circuit using the 74138 to control as many as eight "daughter" circuits, three bits from your output port are used as the address for the "mother" chip to select which "daughter" circuit will be enabled by the output from the 74138. Three more bits are used as the address for the "daughter" chip to select which output latch will be affected. One data bit is used to determine the new value of the selected latch. One more bit is used to enable the 74138. As discussed above, this enable bit must be sent after the other seven bits have settled.

Conclusion

We have discussed how you can use your computer to control or monitor any circuit which contains a simple single-pole, single-throw switch. The hardware and software you will need are quite simple. We have included detailed explanations of the basic circuits plus several complications and embellishments. The rest is up to you. Let your imagination go. Think of anything at all that you would like to monitor or control with your computer. Now you have the knowledge to do it. With a little experimentation, you will soon feel right at home with this sort of interfacing. Good luck.

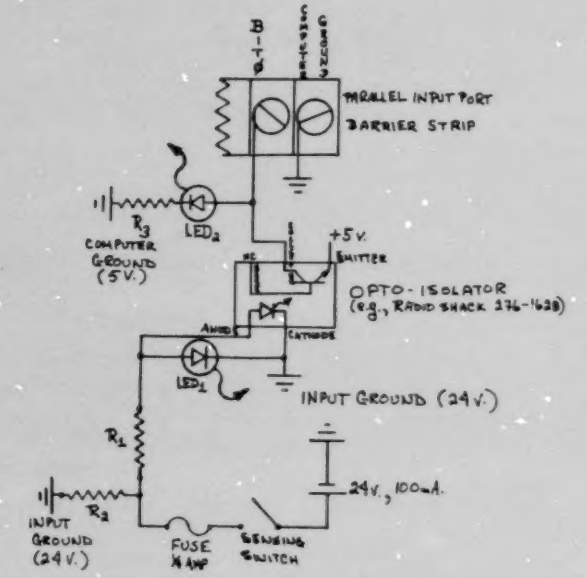


FIGURE 2. NON-INVERTING INPUT CIRCUIT WITH 2 LEDS.

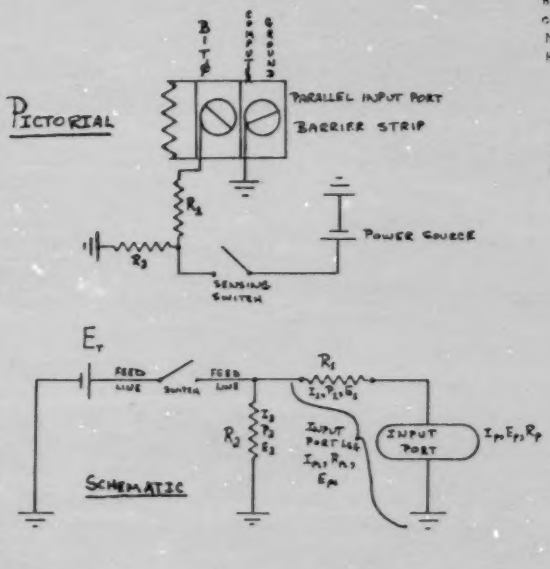


FIGURE 1 GENERALIZED INPUT SENSING CIRCUIT WITH CURRENT LIMITING RESISTORS.

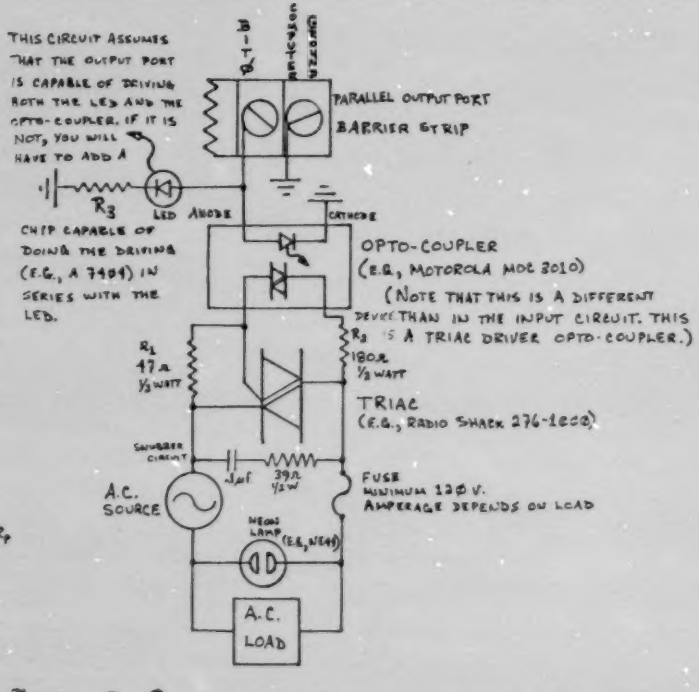
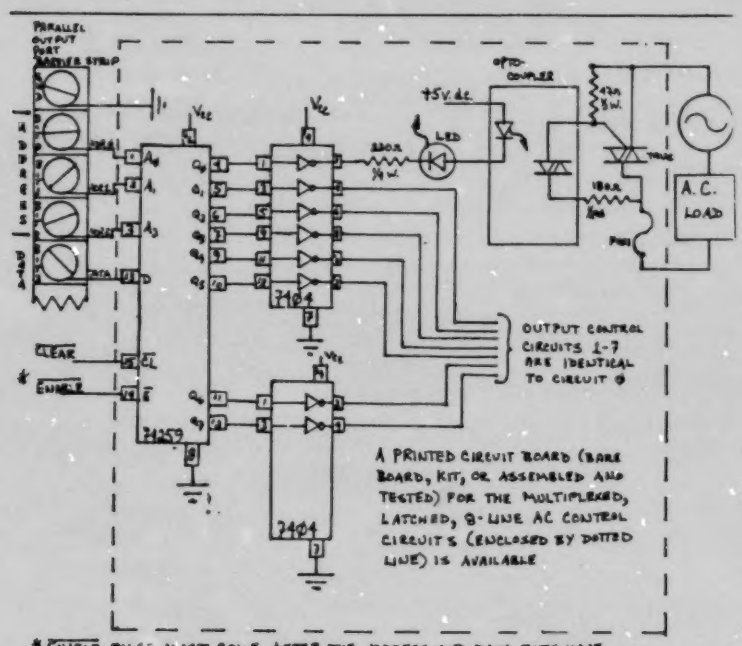
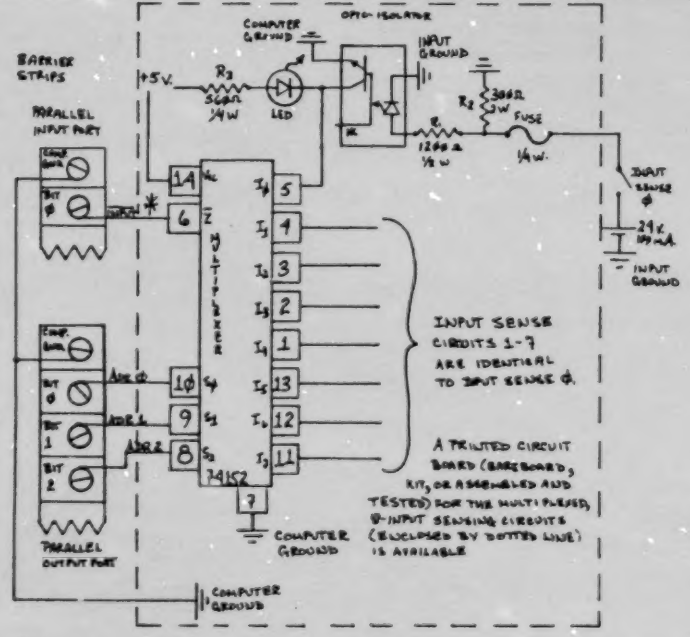

THIS CIRCUIT ASSUMES THAT THE OUTPUT PORT IS CAPABLE OF DRIVING BOTH THE LED AND THE OPTO-COUPLER. IF IT IS NOT, YOU WILL HAVE TO ADD A CHIP CAPABLE OF DOING THE DRIVING (E.G., A 7404) IN SERIES WITH THE LED.
BIT Ø
PARALLEL OUTPUT PORT
BARRIER STRIP
OPTO-COUPLER
(E.G., MOTOROLA MOC 3010)
(NOTE THAT THIS IS A DIFFERENT DEVICE THAN IN THE INPUT CIRCUIT. THIS IS A TRIAC DRIVER OPTO-COUPLER.)
180Ω
TRIAC
(E.G., RADIO SHACK 276-1000)
FUSE
MINIMUM 120V.
AMPERAGE DEPENDS ON LOAD
A.C. SOURCE
A.C. LOAD

FIGURE 3. OUTPUT PORT WITH INDICATOR LAMPS IN FUSED OUTPUT CONTROL CIRCUIT.


PARALLEL OUTPUT PORT BARRIER STRIP
OPTO-COUPLER
+5V. dc.
LED
A.C. LOAD
74174
74259
74174
CLEAR
ENABLE
OUTPUT CONTROL CIRCUITS 1-7 ARE IDENTICAL TO CIRCUIT Φ
A PRINTED CIRCUIT BOARD (BARE BOARD, KIT, OR ASSEMBLED AND TESTED) FOR THE MULTIPLEXED, LATCHED, 8-LINE AC CONTROL CIRCUITS (ENCLOSED BY DOTTED LINE) IS AVAILABLE

*ENABLE PULSE MUST COME AFTER THE ADDRESS AND DATA BITS HAVE BEEN APPLIED TO THE 74259. THIS MAY BE ACCOMPLISHED BY USING A SECOND PARALLEL OUTPUT PORT POKED AFTER THE FIRST ONE. OR YOU MAY USE ONE OF THE LINES ON THE COMPUTER'S CONTROL BUS, E.G., A WRITE STROBE LINE.

FIGURE 4. OUTPUT PORT WITH EIGHT MULTIPLEXED, LATCHED AC CONTROL CIRCUITS


OPTO-ISOLATOR
COMPUTER GROUND
INPUT GROUND
BARRIER STRIPS
+5V.
LED
FUSE
PARALLEL INPUT PORT
74152
INPUT SENSE CIRCUITS 1-7 ARE IDENTICAL TO INPUT SENSE Φ.
A PRINTED CIRCUIT BOARD (BARE BOARD, KIT, OR ASSEMBLED AND TESTED) FOR THE MULTIPLEXED, 8-INPUT SENSING CIRCUITS (ENCLOSED BY DOTTED LINE) IS AVAILABLE
PARALLEL OUTPUT PORT
COMPUTER GROUND

*NOTE WELL THAT THE INPUT IS INVERTED BY THE 74152. THIS MEANS THAT A Φ INDICATES THAT A SWITCH IS CLOSED.

FIGURE 5. MULTIPLEXED INVERTED INPUT CIRCUITS.

BEST COPY AVAILABLE

DELPHI: A Computer Program Featuring Dialogue Among Students and Between Students and the Instructor.

Jerome H. Woolpy

Earlham College

ABSTRACT

Developed over the past eight years with the support of the National Science Foundation, the Exxon Education Foundation, and the Sloan Foundation, DELPHI is now operating at colleges and universities in both the United States and Canada. It serves as a technique for teaching a wide variety of subjects in the social and natural sciences and in the humanities.

* DELPHI fosters inquiry by having students enter questions for other students or the instructor to answer.
* DELPHI supports concept development and writing skills by having students answer questions in essay form.
* DELPHI promotes critical analysis by having students evaluate each other's answers.
* DELPHI encourages dialogue by providing a vehicle for students to discuss each other's comments.

The program allows a student or the instructor to enter questions and provides them for another student or the instructor to answer. These question-answer pairs are then presented to other students or the instructor for comment and evaluation. All text is presented anonymously, so that its substance can be addressed unaffected by non-verbal communication, the identity and reputation of the author, or the relationship of the author to the reader.

Unlike other computer-assisted instruction, DELPHI can be adapted for use in almost any course in a matter of hours. Neither instructors nor students require previous knowledge of computers. The program runs on a variety of minicomputers and the IBM PC microcomputer.

In some courses DELPHI supplements small group discussions; in others it augments essay assignments. Some instructors even use it to replace in-class examinations.

Of the thousands of students and over one hundred faculty who have worked with the program, most evaluate it favorably and elect to use it again. One negative comment about DELPHI from some students is that it depersonalizes learning. Although it would be possible to employ the program in a highly personalized way (by identifying the authors of all text), we have found anonymity to be more of an asset than a liability in most applications; and, personal interaction is available through traditional classroom instruction which DELPHI is intended to supplement.

Some faculty have objected to the excessive time commitment which they judge the program requires, but few deny its significance as an important educational device and a unique and appropriate application of computers to education.

INTRODUCTION

When students become teachers, they often find that they learn more rapidly and have a better understanding of what they learn than they did when they were students.

There are various explanations for this phenomenon of the teaching and learning process. Whatever the cause may be, however, repeated observation of the effect has led us to invent devices which encourage students to behave like teachers, in the hope that they will learn more effectively.

My students first reported that they learned significantly from thinking as teachers when they were invited to grade each other's genetics examinations. They said they could understand the problems better and see the strengths and weaknesses of the various problem solving approaches used by their classmates. This led to a more systematic perspective on the subject matter and clearer a rationale for problem solving.

Based on this preliminary finding, I encouraged my students to behave like teachers in another way. I had them prepare exam questions on material they studied. Exams were then composed from their best questions plus a few of mine. Students knew that their questions would not be chosen unless they were well written and well thought out, and they discovered that by preparing good questions they learned a great deal about the material.

The nature of the exams didn't change markedly, but the students' attitude did. They began to take a more active role in reading their textbook and in listening to my lectures. They began to think about the most important aspects of what they were learning, and about what they ought to get from their own education.

The method worked, but when I used it in classes as large as thirty students, my clerical abilities were taxed to the their limits and I was discouraged from giving examinations as frequently as I might have otherwise.

Fortunately, a colleague and subsequent co-developer of the DELPHI program, Richard R. Johnson (then a professor at Earlham College and now Director of Research at the Exxon Education Foundation) observed that my techniques could be adapted to the computer.

With a computer, clerical jobs could be handled more efficiently and questions, answers, evaluation, and discussion could take place on a more frequent and regular basis. Penmanship problems could be eliminated, all text could be presented anonymously without any special efforts at coding, and the instructor could monitor and participate in the process with much less effort than had been possible with my manual system.

Moreover, there would be no limit on the class size which could employ the technique, nor would it be necessary to take class time to administer or grade exams.

Perhaps most significant, each student would be provided with all the answers to all the questions, and with all the evaluations of the answers.

In 1976 the first version of DELPHI was initiated at Earlham College. During the next two years it was used in eight different courses by seven different instructors. In 1979 we started to disseminate it to other colleges and universities, and in 1983 we modified the program to run on IBM PC microcomputers. (As of July of 1983, DELPHI has been made available to the following schools: Alma College, Austin College, College of St. Catherine, Earlham College, Fairleigh Dickinson University, Franklin College, Miami University, the University of Dayton, and the University of Calgary. It has been applied to the teaching of courses in biology, education, foreign languages, geology, history, law, literature, mathematics, philosophy, psychology, sociology, and writing.

PROGRAM OPERATION

The DELPHI program works in the following way: Each user is assigned an identifying password through which the program keeps track of that user's text entries. When the user signs into the program, DELPHI asks "What do you want to do?"

```
Q=Ask a question
A=Answer a question
C=Comment on an answer
R=Review previously entered material
D=Discuss a comment
S=See a record of the work you've
  done, the assignments,
  and your grades
G=Gripe; Indicate a problem you're
  having
I=Obtain help (more information on
  DELPHI)
E=End the session; sign off DELPHI
T=Take a test
```

The user may respond by entering Q, A, C, R, D, S, G, I, E, or T, and the program proceeds accordingly.

Suppose a Q is entered, indicating that the user wants to ask a question. The program asks the user to choose the category of the question, which is selected from a list prepared previously by the instructor (e.g., Plato, Aristotle, or Kant; or Chapter 1, Chapter 2,...Chapter n; or Week 1, Week 2, ...Week n). The user is then asked to enter a question, as follows:

"Please enter your question, you have a maximum of 10 lines."

(The number of lines available is determined by the instructor.)

The user enters the text of a question, edits the entry, and finally determines that it should be stored in the question text file.

Suppose an A is entered, indicating that the user wants to answer a question. The program asks the user to select the category from which the question is to come (e.g., Plato, Aristotle, or Kant; or Chapter 1, 2,...n; or Week 1, 2,...n). The user is given a question and prompted to answer it as follows:
"Please enter your answer, you have a maximum of 15 lines."
(The number of lines available is determined by the instructor, as is the option to permit the user not to answer the question.) After entering the answer, the user may be asked to rate the quality of the question. The rating can then become part of the evaluation of the user who entered the question.

Suppose a C is entered, indicating that the user wants to comment on an answer. The program selects a question and an answer to that question for the user's commentary. (Answers to questions which the commentator asked may or may not be selected depending on criteria set by the instructor.) The commentator is prompted as follows:
"Please enter your comment. You have a maximum of 15 lines."
(The number of lines available is determined by the instructor, as is the option to permit the user not to comment on the answer.) After entering the comment, the user may be asked to rate the quality of the answer, and the rating can become part of the evaluation of the user who entered the answer.

Besides the question (Q), answer (A), and comment (C) modes, the user may elect to review (R) previously entered material. In that case the following menu appears:
"What do you want to review?"

Q=Your questions with all answers, comments and discussion
A=Your answers with all comments and discussion
C=Your comments with all discussion
O=Other user's questions (including your own)
S=A specific item of text that you know the number of
E=Exit from review.

The user may elect to append discussion (D) to a comment, to continue the dialogue among the questioner, answerer, and commentator, or to protest the rating given by a commentator to the user's answer.

The user may elect to see (S) a record of the work that the user has completed, the assignments made, and the grades given by the instructor.

The user may elect to gripe (G), anonymously or not. This problem or comment is then sent by the program to the instructor.

Finally, the user may obtain more information about DELPHI by entering I, sign off the system by entering E, or take a specially designed test (T) (consisting of questions selected at random by the program from a list of questions flagged by the instructor).

The instructor can also take advantage of several auxiliary programs, which include the following functions:

1) Setting up and modifying the parameters of a course, such as the names of the categories, the number of lines allowed for questions, answers, and comments, and the option to permit users not to have to answer particular questions.
2) Sending messages to individual users and groups of users.
3) Accessing text files and providing printouts, by user, by question, or by category.
4) Deleting text files.
5) Setting deadlines for the number of questions, answers, or comments which must be entered before a certain date.
6) Setting prerequisite assignments which must be completed before entry is allowed in certain modes or categories.
7) Assigning letter grades to each user for work done in each category.

APPLICATIONS

1) In a course on Animal Behavior, DELPHI has replaced all written examinations. Students enter one or more questions on each week's work, enter one or more answers to each other's questions, and enter at least three comments to each other's answers. Their grades are determined, in part, by the ratings that they give to each other's questions and answers. They spend from one to three hours per week at computer terminals.

2) In an introductory course on Human Biology, instructors enter questions each week and students must each answer three questions which the program selects at random. Advanced students in an upper level course comment on and evaluate the answers given by the introductory students. Thus the same files enhance the educational process at two different levels.

3) In an interdisciplinary seminar organized by senior undergraduates with different majors, students answer each other's questions. Then, depending on the subject matter of a particular week's work, various instructors from different

departments comment on and evaluate the answers. Because of DELPHI, such courses can be offered without requiring the full time instruction of one or more teachers.

4) In a large introductory course in political science, with approximately one hundred students and only one instructor, regular small group discussions are not possible. However, the use of DELPHI has provided one or more hours of written discussion per week for every student.

5) In a freshmen composition class, students read each other's weekly papers and DELPHI allows each reader to offer commentary to each author. This prepares them for face to face critique sessions, improves their critical skills, and permits the instructor to assess each student's ability to determine other student's writing problems.

6) In a geology class, students are each asked to "describe the numbered rock that was assigned you from the collection in the bag at the side of the computer terminal." The answers consist of formal geological descriptions. The comments, also entered by students, evaluate the descriptions and help determine whether students can tell which rock was described.

7) In a sociology class, students from various backgrounds discuss the sensitive issue of what it is like to be a member of a minority group in a class consisting mostly of white middle class students. Issues are confronted which are never dealt with directly in a conventional discussion, where the participants must address each other directly.

8) In a class on psychological counseling, students are asked what they would do with a particular case or in a particular situation which might arise in counseling. Alternative answers to the same case are provided by different students, and comments about these answers are made by other students and the instructor.

9) In a class in French, students get one to three hours per week of practice in writing French dialogue. The instructor comments on and evaluates each student's work each week.

DISCUSSION

In all these applications of DELPHI, users must express themselves clearly enough to communicate particular ideas, concepts, and feelings. Gestures are not possible, the identity of the author cannot convey meaning, and the author cannot rely on a later opportunity to clarify intended meaning. As a result, students learn to express themselves in writing.

Unlike most other computer assisted instruction, DELPHI permits users to deal with the complexity and subtlety of subject matter. Though it certainly provides plenty of drill and practice, it is by no means limited to matching questions, multiple choice, or other structured learning and testing devices. And, unlike most other devices, it makes the student both a resource in the learning process and a responsible judge of what is important and what is good.

EVALUATION

Instructors who use DELPHI report an improvement in the writing abilities of their students. Students understand complex concepts more rapidly and in greater depth than they did in the same classes before DELPHI was adopted. There is also a marked improvement in the quality of student discussion following repeated experiences with the program over a period of several weeks. Many instructors elect to use the program again.

Although a few instructors have objected to what they judge to be the lengthy task of reading the entire DELPHI dialogue each week, we have noted that the most significant advantage of the program as a teaching device comes in its actual use by students in student to student dialogue. Therefore, it is not essential nor even recommended that instructors read all the text on a regular basis. Indeed, reading particular items in response to specific student requests is usually sufficient to obtain positive results from the program.

Students consistently report that they learn more from using the program than they do from other, more conventional approaches. Many elect a course, at least in part, because it employs DELPHI. Although a few have indicated that they will not select courses which use the program, they normally justify the decision by stating they don't want to think that hard, work that regularly, or learn that much about certain subjects. Students also raise questions about their ability to evaluate each other's work or have their work evaluated by other students. Still, while student evaluation is not essential to DELPHI's application, experience supports our contention that it contributes significantly to the effectiveness of the program. Peer evaluation increases student awareness of each others' and their own strengths and weaknesses, de-emphasizes the role of the

instructor as evaluator, and provides practice in communicating criticism.

Formal evaluation of DELPHI has consisted of running sections of large team-taught courses, where the test performance of one section which uses the program regularly is compared to another section which does not use the program. So far no diagnostic devices, except the reports of instructors and students indicated above, have revealed any measurable differences between experimental and control groups.

TECHNICAL DESCRIPTION

For minicomputers DELPHI is programmed in FORTRAN IV. It is supplied on nine track tape with 800 or 1,600 bytes per inch in ASCII or EBCDIC. To support approximately 100 students each using the program one to three hours per week and retaining all text on line for an entire semester, we estimate about 44,000 bytes of core and ten megabytes of disk are required. Removing text from previous weeks will provide substantial savings of disk space.

For microcomputers DELPHI has been translated into Microsoft FORTRAN. A version for the IBM PC and IBM XT running under DOS 2.0 is now available. The IBM microcomputers must be equipped with at least 192 kilobytes of random access memory. The programs are supplied on 5.25 inch diskettes.

DEVELOPMENT OF A BASE FOR COMPUTER ASSISTED LEARNING METHODS

HANAN YANIV

This article will attempt to present the benefits of individualized learning , and the computer as the most efficient of such systems. The reader will be presented with the considerations to be taken into account when choosing software and equipment; when building the organizational structure; with the method for preparation of the computerized lessons; and with overall construction of the system.

Note: Any mention of learning in this article connotates solely the impartation of knowledge. Impartation of knowledge may be described as: presentation of concepts, processes, ways of thinking, techniques for memorization and decision making. There is absolutely no intention to present the computer as an educating tool.

INDIVIDUAL LEARNING IS THE MOST EFFECTIVE TEACHING FORUM.

In order to comprehend this absolute determination, it is necessary that we consider the alternative, classroom learning. The scenario is familiar to all: the teacher stands at the blackboard, in front of the class, faced by several 'tens' of students.

Learning ability of the students usually is divided thus:

Theoretically, the teaching process is said to be directed at the average students, however, is this true in practice? Clearly the weaker students can not keep pace and therefore ask clarifiying questions (in the best case). The teacher, who must pay attention to the difficulties encountered by his students begins to add explanations not required by the majority of the class. In the worst case, the weaker students invisibly 'drop out' and despite their continued presence in the classroom, either merely understand nothing, or become disturbances to the continuation of the lesson. And as for the good students? They become bored, can't find themselves, disturb, and "disappear." In such situations the teacher quickly becomes frustrated and frazzled.

Without taking into account the previous considerations, the teacher is not the most effective teaching method. The percentage of relevancy in what is heard in the classroom is usually quite low.

Teachers are subject to moods, frustration, bitterness, and tiredness, which undoubtedly influence their students. In addition, the average teacher (many teachers will have to forgive me) tends to be boring, thus limiting the motivation of his students, and of course: it is not technically feasible that a teacher give each and every student reinforcement for each and every positive activity.

BEST COPY AVAILABLE

Individualized learning systems were developed in answer to this gloomy picture.

According to the principals of individualized learning, each student advances at his own pace, without fearing erring, or fearing displaying his need for additional explanations. In fact all of the above mentioned restrictions as regards classroom learning are eliminated in individualized learning. I again remind those who fear the loss of the socializing experience, that we are discussing impartation of knowledge solely here, and the significantly important task of educating remains that of education. I have no doubt that any educational activity will leave wide room for social interaction.

Perhaps it would be worthwhile to explain briefly here what exactly individualized learning is: this is an approach in which the student is given the tools with which to teach himself. What are the tools: usually a booklet, or a compilation of booklets and other teaching aids. For example: a system for teaching of topography combines booklets, maps, pictures, and plastic models of landscapes. The booklets are designed in such fashion that the student is provided an explanation and then questionned regarding it. If he correctly comprehends the material, he is referred to pages in the booklet on which the lesson continues, if he fails, he is directed to pages on which he receives additional explanatory materials.

THE COMPUTER IS THE MOST EFFICIENT FORM OF INDIDIVIDUALIZED TEACHING.

For a booklet to include a large number of explanatory tracks, and illustrations for as many of the various levels of comprehension as possible, it would have to be quite massive, making the teaching process most complex. It is also difficult to illustrate dynamic developments with this inanimate instrument. The computer facilitates display of many learning tracks, and dynamic display of occurrences (with the aid of animation). The computer supervises the learning process by forcing the student to be involved at all times. A student not paying attention who errs, lengthens his own study time as he will be forced to receive additional explanations. The computer also kindles motivation to learn due to the interaction between the student and the computer which stimulates the student to continue.

To this point we have seen how effective a teaching instrument the computer is.

Its humane characteristics must also be noted, as it has endless patience for the weaker students, it also eliminates their having to display their limitations in public, and reinforces their every achievement, encouraging them to continue. A final point: the computer facilitates updated and efficient renewal of the teaching materials, enabling presentation of the latest developments.

WHAT CAN BE DONE WITH COMPUTERS IN THE TEACHING SYSTEMS?

Testing - Testing the knowledge level by computer makes possible immediate receipt of results and adaptation of the questions to the level of the student.

Drill & Practice - The most accepted usage of computers in today's teaching systems. Enables review of material by drill. Especially frequent in language and mathematic studies.

C.A.L.- Computer Assisted Learning - i.e. presentation of new materials to the student. Such sytems generally include some of the previous characteristics, for testing and drilling of the subject matter taught.

Simulation - With the assistance of the computer it is possible to simulate processes and to imitate reality, in many cases saving money or needless dangers.

Development of Skills - Techniques for memory, thinking, and decision making can also be conveyed via computer by exercises and immediate feedback.

Possible applications of the teaching system are, of course, dependent upon the type of equipment. Table No. 1 will present the technical requirements of the computers for every type of application.

We will relate to the following areas:

Internal memory: ability of the computer to process a large quantity of data simultaneously.

External memory: ability to connect to a Winchester disk of large capacity - ability to work with a large lessons library and ability to preserve students files for statistical analyses.

Communications network: a communications network between the computers of the students and a central memory which will enable feeding of results into files of students, preserved on the disk, and use of the lessons Library in a modular fashion facilitating constructing of a lesson suited to the needs of the individual.

Graphics: ability to present drawings and dynamic presentations on the computer screen to make the illustration process clear.

Color: ability to present graphics and headlines of varied colors, for improved illustrational ability and improved learning and memory.

Sound: for the addition of sound effects.

Video Disk Connection: Ability to control a video disk enabling random presentation of video pictures on the screen and the possibility of overlaying computer graphics on the video picture.

Input devices: (touch screen, light pen, joystick, etc.): for facilitation of interaction without use of the keyboard.

Authoring language: Elite programming language, to facilitate programming by the unskilled developer.

TABLE NUMBER 1

CRITERIA FOR CHOICE OF EQUIPMENT ACCORDING TO APPLICATIONS

	Internal Memory	External Memory	Comm. System	Graph-ics	Color	Snd	Video Disk	React. means	Link lang.
Testing	Small	-	*	?	?	-	-	*	?
Drill & Practice	Small	-	-	?	?	-	-	*	?
C.A.L.	Med.	*	*	*	*	*	*	*	*
Simulation	Lg.	*	-	*	*	?	*	*	?
Development of Skills	Sml to Med	-	-	*	*	?	-	*	?

Key: * = Vital ? = Depends upon application - = Not required.

BEST COPY AVAILABLE

WHAT ARE THE ORGANIZATIONAL CONSIDERATIONS TO BE TAKEN INTO ACCOUNT?

Although it is true that according to the model about to be presented any educational set-up could feasibly design its own C.A.L. system, and to prepare the teaching materials with the aid of its own teaching staff, the task is not a simple one.

First of all, didactic considerations must be taken into account. There is no point in feeding existing lesson structures into the computer (even if they were designed for individualized instruction). Rather the computer should be exploited in the areas in which it is most able to contribute, such as the building of tracks, illustration, feedback, and tests.

To this end, skilled potential must be brought to the system, which will serve as guide and assistance to the instructors who will write the lessons. This function should be filled by a person who has completed educational program writing, or educational counseling with a specialization in individualized learning.

Of course, the teachers will serve as the source for the content of the lessons.

This is not a recommendation based upon economic considerations. Our leading theory recognizes the instructor as the best source of content because he is familiar with the means of illustration most suitable to each type of student, and because of his familiarity with the means of testing achievement of instruction goals. An additional consideration relates to the desire to imbue every lesson with the spirit of the teacher who wrote it, in order to create a set of different lessons which vary one from the other in the humor in them, in the means of reinforcement in them, and in the means of illustration they contain. This is the means of leaving a bit of humanness in the teaching machine.

Use of instructors as a source of content will also open before instructors an alternative means of employment, facilitating acceptance of the system without suspicion and alienation.

The Software: This usually creates a picture of a computer room filled with professional programmers with gigantic salaries, etc. To calm you: There are elite programming languages designed for the development of educational programs. These languages facilitate the training of people within a period of approximately one month without previous training. Thus it is perfectly feasible that instructors and students be taken and retrained in use of these languages.

Thus, the staff described to this point is composed of a didactic consultant, a contents expert, and a programmer (another instructor or a student).). It would seem preferable not to give the instructor writing the lesson the task of programming it, due to the human tendency to take short cuts which could negatively effect the quality of the lesson. This could occur if the writer knows that he will later have to program his ideas.

HOW LESSONS ARE WRITTEN:

Here lies the most problematic part of the entire system. It must be recalled that conventional instruction material (however good it may be) is not suitable for individualized learning, especially of the computerized sort. The process of preparation of the a lesson for computerization occurs in two stages: dismantling and assembling.

The purpose of the dismantling is breaking subject matter into its most primitive concepts to be reassembled in associative tracks.

These tracks present associative connections between the various components which will enable sealing of the learned material in the brain of the student over a period of time. During the building of the tracks, consideration must be given as well to testing, reinforcement, and summarizing. The tracks are then presented in flow-chart form.

This instrument is employed for two purposes. First of all, it gives an overview of the lesson, and presents the integration between the various tracks. On the other hand, this is an important instrument for the programmer, according to which he will build the lesson. Of course in coordination with the flow-chart the screens of the lesson are prepared, together with the graphics, the questions, and the other components.

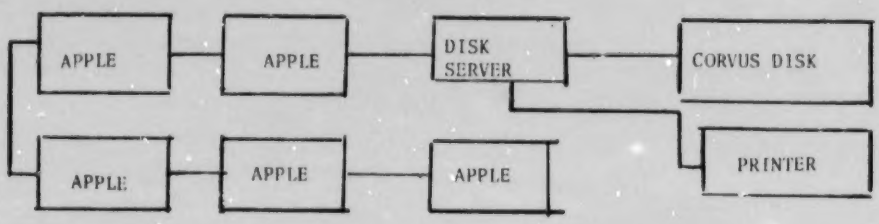
In addition it is worthwhile to use an instrument we call the "concept dictionary." The concept dictionary presents the student with all the concepts he has learned to this point in the lesson and offers him the possible of receiving additional information on each and every concept, according to his choice. This is an important instrument for organizing for the student the material he has learned, and defining for him the concepts he must know. It is also important in that it gives the student the opportunity to ask questions regarding unclear concepts and to receive explanations in addition to those he has already received and not properly understood.

BEST COPY AVAILABLE

STRUCTURE OF THE COMPUTERIZED CLASSROOM

Following is a chart presenting the structure of the classroom, which answers most of the criteria presented in Table No. 1.

This chart presents the system we built. The goals we set ourselves centered on the third criteria in Table No. 1 - C. A. L. Accordingly we defined our requirements and have chosen the following:



No. of computers is determend by the specific needs.

THE LANGUAGE: APPLE SUPER PILOT.

This is one of the best Authoring languages available. It is based on PILOT which is a veteran language in the area of educational development but is quite restricted. The SUPER PILOT added layers of graphics, animation, color, and sound possibilities to the language, as well as the possibility of varied programming. The most significant benefit is the ease of use. It is easily taught, (even to those with no previous background in computers) within a month. The final product generally answers all the needs of the lesson planner.

The process of planning the lesson centers around use of 4 friendly editing programs:

1. lesson text editor
2. graphics editor
3. character set editor
4. sound editor

Every editor is preceded by a help screen which includes a summary of the possibile actions. The product is stored in a file which is easily called for and used. Creation of colored drawings and combination of the text with them, movement of objects, boldfacing of letters, enlargements, coloring, are all executed by just a few commands.

To aid the programmer, there is a graphics tablet which facilitates the copying of drawings from paper with the aid of an electronic pen, their coloring, their enlargement or reduction, their movement, etc. Creation of a drawing, no matter how complex it may be, is merely a matter of minutes.

The language permits work in Hebrew and support of all the input means which I will presently be discussing.

An additional important detail is the system for management of the teaching which facilitates statistic processing of the data on items, students' and classes' achievements, as well as their presentation in a graphic, printed manner (SUPER PILOT LOG).

THE COMPUTER:

From the language's name, it must be clear to the reader that the computer to be employed is the APPLE (or its compatible). The required amount of memory is 64K, which will require users of the APPLE II+ (or its equivalent) to add 16K of RAM memory. Users of the APPLE IIe will, of course, not need to add this. A colored monitor will be attached to every computer.

INPUT MEANS:

In this era means of input include a wide variety of tools: Joy Stick, light pen, mouse, touch board, and others. The last, the touch board, only recently came out and presents a cheap and efficient means of involving the student in the lesson with the aid of his finger. A board of approximately 25 x 25 cm² is laid on the table. By touching it, the student is able to note a place on the screen. It appears that this solution (the Koala touch pad) answers the need to find a comfortable means of interaction between the student and the computer.

BEST COPY AVAILABLE

THE CLASSROOM STRUCTURE:

All the computers are connected to the disk. The disk is in fact a great memory reservoir in which the students' files are stored next to the teaching materials. The student receives his lessons on his computer screen and feeds back his responses and answers. This data is analyzed by the management system which receives a picture of the situation of the student, the lesson, and the class. The large memory reservoir permits storage of concept libraries and modular form compilation of special lesson plans for each student in accordance with his scholastic level and fields of interest. We employ a CORVUS disk and the network connecting the disk to the computers (also of the CORVUS family) is called OMNINET.

It is also possible to connect a printer to the system and to print portions of the lesson, such as the interim summaries, concept lists, graphic summaries, or division of grades, students' printouts etc.

CONCLUSION:

Taking into consideration the huge quantity of information required for professional specialization in our times, and taking technological development considerations into account, there is a definate need today - and the possibility exists - for erecting sophisticated, efficient teaching systems.

The outlines presented in this article suggest means of relating to the organizational, theoretical, and technical areas. Undoubtedly this is not a users' guide...(MANUAL), but can aid in the characterization process prior to erection of the system.

A functioning system such as that presented has been designed and implemented by the writer (with his team including Ariella Perlmutter, Anat Adler, Dudi Holzman, and Tzachi Segal) who find its achievements most encouraging.

EDUCATIONAL MUSIC SYSTEMS FOR MICRCCOMPUTERS

Susan Jenkins

The Ohio State University

INTRODUCTION

In this presentation we will be concerned with off-the-shelf sound systems that can be used for music education. Uses, criteria, and price ranges of sample systems will be examined.

Music and sound capabilities on micros were considered frills until recently. Sound capability was often known to many as one of the micro's many "bells and frills." Even in the educational environment, use of sound was developed at a much later date than other capabilities were. However, it is possible to seriously use the sound capabilities of many micro systems for music education.

USES FOR A MICRO SOUND SYSTEM

Micros offer a wide range of educational programs for use in the classroom or individually. Some of the uses for micro systems would include:

* Drill and practice
* Music dictation and reading
* Developing aural listening skills
* Research in music education
* Teaching music composition

CRITERIA FOR A GOOD SOUND SYSTEM

Many of the criteria that one looks for in a good sound system include those one finds on any well-written computer instructional system. These include an interactive approach, good instructions, a balance between student and program control, and a logical organization of presentation. A good system also should have good screen design, a well-paced flow of exercise questions-responses, and accurate methods of evaluating student responses.

Specifically sound oriented criteria include use of interesting sounds that show variety and imagination. Examples from literature should often be used for emphasis and clarification. Sound programs should show more than a rudimentary knowledge of music theory, perception, notation and acoustics on the part of their authors. The program should in addition, take fullest advantage of what the micro is capable of doing with sound.

WHAT YOU CAN EXPECT

The main categories to investigate when examining a micro sound system concern the areas of control, flexibility and ease of use. Music educational programs, for the most part, in particular the ones written for Apples, require an additional hard board.

Criteria that are hardware dependent include:

* waveform - usually some form of a square wave, harmonically generated
* voices - from 1 to 4 voices, maximum of 16
* range of frequency - usually 4-8 octaves

* range of amplitude - depending on amplification system and board used

Criteria that are software dependent include:

* envelope control - attack, decay, and sustaining time
* filtering capabilities
* timbre control - ability to change the sound quality, or to program the sound quality
* change of tempo
* modulation capabilities - tremolo and vibrato are included here

Most sound programs are usually devoted to one music subject or area. You can expect the sound programs that run on micros to try to cover one of these areas:

* Music Theory
  - Harmony
  - Key Signature recognition
  - Visual and Aural recognition of chords and chord groupings
* Ear Training
  - Dictation - melodic, harmonic, and rhythmic
  - Chord Recognition
  - Solfeggio and scale degree drill
  - Mode recognition
* Music Appreciation
  - Composers' works by period
  - Music symbols
  - Music terminology
  - Instrumert names and ranges
* Music Composition
* Instrumental instruction (private or group)
  - Guitar
  - Percussion
  - Melody instruments of the middle range in pitch

SOME SAMPLE SYSTEMS

Name:	Sebastian of Micro Music
Manufacturer/ Vendor:	Temporal Acuity Products, Inc.
Hardware Required:	Apple II+ with 48K RAM, disk drive, video monitor, MMI DACBOARD ($125)
Price Format:	Sebastian diskette $125, Micro Music programs range Music Theory $125-$200; Ear Training $100-$200; Music Appreciation $50-$150; Composition $150-$200; Instrumental instruction $75-$150
Audience:	Students of all ages; students and teachers with no prior computer experience

Name:	Basic Ear Training, from ALF
Manufacturer/ Vendor:	Peripherals Plus
Hardware Required:	Apple II+ with 48K RAM, disk drive, video monitor, ALF Card
Price Format:	Disk of four programs $50; all ALF software and board $198
Audience:	Students of all ages; students and teachers with no programming experience

Name:	Music Machine
Manufacture/ Vendor:	Commodore
Hardware Required:	Commodore 64 with 64K RAM, disk drive, video monitor
Price Format:	Diskette $29.95 each; no additional board required
Audience:	Personal computer owners with a strong interest in music; students and teachers with no prior computer experience

EDUCATIONAL MUSIC SYSTEMS FOR MICROCOMPUTERS

Name:	Music Maker
Manufacturer/ Vendor:	Texas Instruments
Hardware Required:	Texas Instruments 99/4A with 16K RAM and cartridge storage, video monitor
Price Format:	Cartridge $39.95 each; no board needed
Audience:	Hobbyists interested in music appreciation; students and teachers with no prior computer experience

LIST OF SAMPLE SYSTEMS

Andex, Sirius Software, Inc., Apple II, $29.95.

Apple Super Music Synthesizer, Applied Engineering, Apple II, $159.

Compu-Music, Poland Corporation, Apple II+ and IIe, TRS-80 I and III, IBM PC, NEC PC-6000 and -8000, cost varies.

Early Games Music, Counterpoint Software, Apple II+ and IIe, $29.95.

Electronic Duet, Insoft, Apple II, $29.95.

Micro Music, Temporal Acuity Products, Inc., collection of software and board, Apple II, $50-$400.

The Music Box, Newtech Computer Systems, TRS-80, $149.

Music Comp, Lyco Computer Marketing, Atari, $33.75.

The Music Card, ALF Products, Inc., Apple II, MC-1, $195; MC-16, $245.

Music Maker, Sublogic Communications Corp., Apple II, II+ and IIe, $49.95.

Music Readiness, Sterling Swift, Apple IIe, $69.95.

Music Synthesis System, Micro Technology Unlimited, Pet, AIM, KIM, Apple, $50-$89.

Orchestra-90, Software Affair, TRS-80 Model III, $149.95.

Piper, Avacus Software, Vic 20, $22.95.

Player Piano, Dunacomp, Atari, $19.95.

Roland CMU-800 and Compu Music, Roland Corp., Apple II, NEC, $465.

Soundchaser Digital, Passport Designs, Inc., Apple II, IIe, $1250.

Synthy 64, Abacus, Commodore 64, $37.95.

Vic Music Composer, Thorn EMI, Vic 20, $39.95.

BIBLIOGRAPHY

Ahl, David H. "School Uses of Microcomputers," Creative Computing, October, 1983. Information from Johns Hopkins survey. October issue of CC contains a number of educationally oriented articles.

Arenson, Michael A. "An Examination of Computer-Based Educational Hardware at Twenty-Eight NCCBMI Member Schools," Journal of Computer-Based Instruction, Aug-Nov, 1978.

Association for the Development of Computer-Based Education. "Computer-Based Instruction: A New Decade," Proceedings of the Annual Conference of the Association for the Development of Computer-Based Instruction, 1980.

Bales, W. Kenton. "A Model for Generative Harmonic Dictation," available on Microfiche.

Chamberlain, Hal. "Musical Applications of Microprocessors," Musical Applications of Microprocessors, 1980.
Tries to include everything; is most useful as a reference guide.

Gross, Dorothy; Griffin, Wendy. "Implementation and Evaluation of a Computer-Assisted Course in Musical Aural Skills," AEDS Journal, spring, 1982.

Hofstetter, Fred T., "Microelectronics and Music Education," Music Educators Journal, April, 1979.

Jenkins, Susan. "Sound Systems on Microcomputers," IEEE Proceedings for National Symposium for Small Computers in the Arts, Autumn, 1983.

Lubar, David. "So You Want to Buy a Music Synthesizer," Creative Computing, Winter, 1982.
Sample systems and brief explanation of some terms.

McLean, Robert S.; McLean, Ruth W., Microcomputers in Higher Education, 1981.
Edited collection of articles.

EDUCATIONAL MUSIC SYSTEMS FOR MICROCOMPUTERS

Peters, G. David. Courseware Development for Micro-Processor Based Instruction in Music, 1979.

Steele, Douglas J.; Wills, Barry L. "Microcomputer-Assisted Instruction for Musical Performance Skills," Technological Horizons in Education, January, 1982.

Tubb, Phillip. "Pick and Choose," Creative Computing, Winter, 1982.
Somewhat biased but informative comparison of ALF, Micro Music and Mountain systems.

ABSTRACT

Music educators have a growing number of programs on microcomputers available to them. Many programs may be used in the classroom and/or for individualized instruction. This paper is concerned with off-the-shelf sound systems for microcomputers which can be used for music education. Uses, criteria, and price ranges of sample systems are examined.

COMPUTER PRESENTATION OF ACOUSTIC SPECTRUM PATTERNS AS AN AID IN TEACHING MUSIC PERFORMANCE SKILLS

Peter Simpson, University of Kentucky

Allen Goodwin, University of Kentucky

ABSTRACT

The feasibility of using a microcomputer for visualization of musical timbre is discussed. Peripheral devices which enable a microcomputer to function as a software controllable real-time spectrum analyzer are described. Examples of visual differences of spectral patterns between certain desirable and undesirable tone qualities are shown for voice, violin, French horn, flute, and bassoon. Possible benefits for teaching music performance skills are discussed.

Introduction

Computers have been used in music education primarily for teaching concepts of music theory. Areas such as ear training or rudiments of notation are particularly well suited to computer instruction, since they typically involve repetition and drill and objectives can be defined precisely. However, there has been little, if any use of computers in teaching music performance. This report explores the feasibility of using computers for the visual display of sound, a process which may aid students in learning to play or sing sounds having (or not having) particular timbres.

A musical performer, as Roederer (1970) has pointed out, is at once both a producer and receiver of sound. Thus the teacher of performance skills must be concerned with training a student's perceptual skills, for they ultimately control the sutdent's sound. A problem for the teacher is communicating his concepts regarding sound to the student. Typically, the teacher uses psychological imagery, which is necessarily vague and subject to various interpretations. For example, a performer may be told that he should try to make a sound which is "brighter" or has more "buzz."

A problem for the student is knowing when the sound production is correct. Until the student's auditory memory has a store of sensory information suitable for monitoring and controlling his sound output, he must rely on the teacher for feedback concerning the correctness of the sound. This problem is compounded in practicing because the student is expected to remember those sensations experienced in the studio under the guidance of the teacher and to replicate that sound production in the practice room with no outside guidance. Moreover, a performer's perception of his own tone is different from that of an audience, and it may take the performer a long time to develop a concept of what the audience hears.

Acoustical Concepts and Spectrum Analysis

A single musical tone consists of several frequency components existing simultaneously. Each frequency component is called a partial, and the collective set of partials constitutes the spectrum of the sound. Sounds having different timbres are characterized by different patterns of energy distribution among the partials. Frequency areas where energy is concentrated are called formants. A visual display of a sound's spectrum is thus a graphic representation of the sound's timbre. Machines capable of

visually displaying a performer's sound could conceivably aid a student in recognizing those aural, tactile, and kinesthetic sensations existing at the moment a sound is produced correctly.

A spectrum analyzer is a device which divides the aural spectrum (20-20,000 Hz.) into several smaller frequency ranges, and displays by means of a graph the amount of energy present in each frequency band. Spectrum analyzers which continuously display information at virtually the same time as a sound is produced are called real-time spectrum analyzers. Peripheral devices and software are available which give microcomputers the capability to function as real-time spectrum analyzers. The use of software control enables the selection of the particular spectral features to be monitored as well as the video display formats.

Fundamental tasks involved in developing practical computer systems for timbre visualization include: (1) identifying the acoustical parameters of sounds which are responsible for their desirable (or undesirable) timbres, and (2) devising visual formats which are effective in displaying this information to the performer.

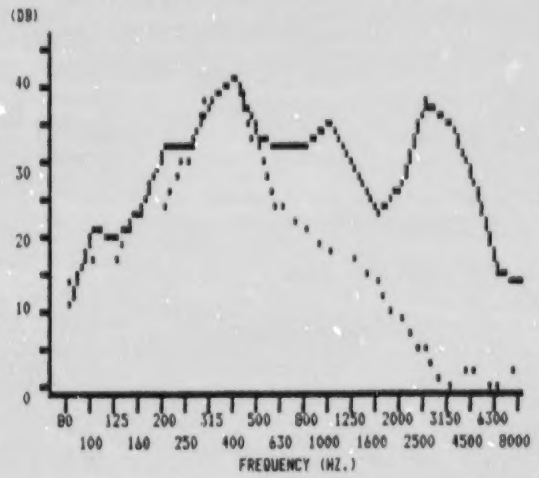
In order to assess the feasibility of the process described above, applied music teachers representing various performance areas were asked to identify certain aspects of tone quality which were typical concerns in their teaching. Sounds having particular good and bad tonal characteristics were then produced. A spectrum analysis of the tones was performed using an Eventide Model AIB-232 one-third-octave spectrum analyzer, interfaced to an Apple IIe microcomputer. With this equipment each frequency band covered a frequency range one-third octave wide. While the display did not provide sufficient detail to assess the strength of each partial in the sound spectrum, it did indicate the general pattern of energy distribution across the sound spectrum. These spectral displays demonstrated the functional capability to illustrate whatever tonal qualities may be of interest to an individual music teacher.

The following discussion explains how timbre visualization via a microcomputer display could be used in the teaching of voice, strings, brass, and woodwind performance.

Voice

In sung tones there are typically two formant peaks in the range 300-2300 Hz. The frequencies of these formants vary with different vowels. For vocal sounds which have an "operatic" or "ringing" quality, there is an added formant in the range 2700-3200 Hz. This formant is termed the singer's formant (Sundberg, 1977). Voice teachers seek to develop the singer's formant because it gives the singer an advantage in being heard over an orchestra.

Figure 1. Spectrum graphs of two tones produced on the vowel /u/ and the pitch F3 (175 Hz.) by a baritone singing. The solid line represents a sound having a strong singer's formant. The dotted line represents a sound having a dark, "throaty" quality.



The spectrum graphs in Fiqure 1 show clear differences between tones produced on the same vowel with and without the singer's formant. The solid line represents the spectral pattern for the vowel /u/ as in "food," produced by a baritone with a dark, "throaty" quality. A strong first formant is seen near 400 Hz., a weaker formant near 800 Hz., and a sharp decline in energy above 1600 Hz. The sound produced with a strong "ringing" tone had a strong first formant near 400 Hz., a strong second formant near 1000 Hz., and a prominant third formant near 2800 Hz. The visual differences of the spectral patterns for the two tones described above would be readily apparent to a singer attempting to follow a teacher's directives for producing the desired formant. The

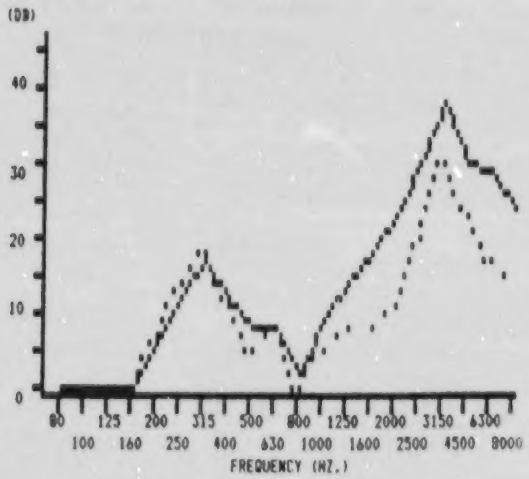
value of visual feedback in developing the singer's formant has been demonstrated in an earlier study (Goodwin, 1982).

Violin

A good violinist is expected to be capable of producing a range of timbres, including "centered" and uncentered" tones. This is achieved by an optimum bowing pressure, bowing speed, and distance from the bridge.

Figure 2 indicates that the spectrum of a centered tone, in contrast with an uncentered tone, was characterized by a higher level of energy in the range 1000-6300 Hz., with a strong peak at 3150 Hz. Since this corresponds to the frequency range where the ear is most sensitive, centered tones are perceived as subjectively louder. The spectral differences are clearly visible on the spectrum graphs, as they would be to a student violinist learning to produce a centered tone.

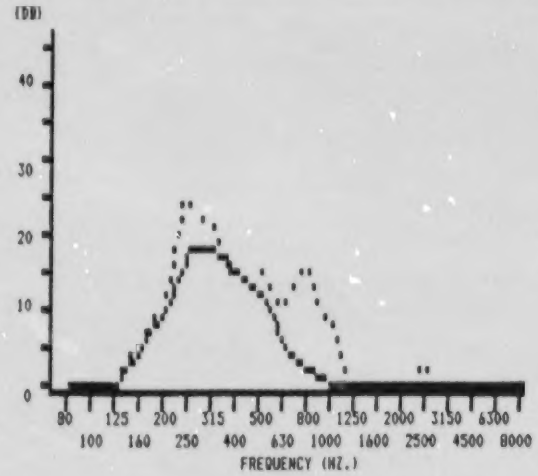
Figure 2. Spectrum graphs of two tones produced on a violin on the pitch D4 (294 Hz.). The "centered" tone is represented by the solid line, and the "uncentered" tone is represented by the dotted line.



French Horn

A typical problem of beginning horn players is using too much lip tension. This results in a tone often described as "shallow" or "tinny." Figure 3 shows that the spectrum of such a tone had prominent peaks near 600 Hz. and 1000 Hz. in addition to a strong fundamental frequency. In contrast, the well produced tone had a strong fundamental frequency near 250 Hz. with a gradually tapering decline in energy to 1000 Hz. Thus timbre differences of this type could be clearly visualized on the video display.

Figure 3. Spectrum graphs of two tones produced on a French horn on the pitch C4 (261 Hz.). The tone produced normally is represented by the solid line, and the tone produced with tight lips is represented by the dotted line.



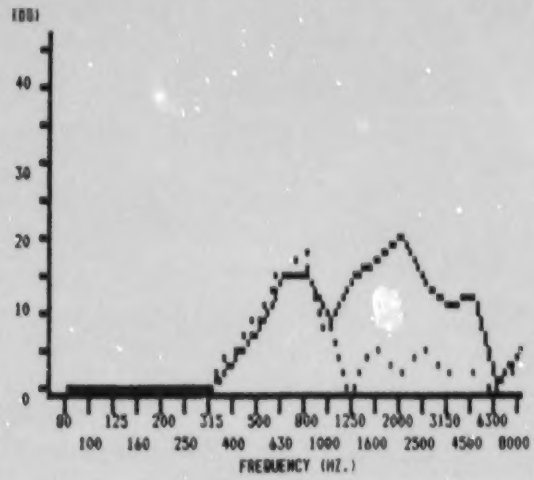
Flute

Flute tone is, perhaps, a good candidate for the use of timbre visualization because the player exercises direct control over the instrument's sound. The terms "centered" and "uncentered" are also used frequently in flute pedagogy. There are various factors influencing the production of centered tones, including the angle of the air stream against the tone hole.

Figure 4 shows clear differences in the spectra of tones with different angles of the air stream. When air was blown across the tone hole, an uncentered tone was produced. The spectrum included a strong fundamental frequency and weak upper partials. When air was directed more into the tone hole, a centered tone was produced. The spectrum of the centered tone included a strong fundamental frequency in combination with strong upper partials

in the range 1500-3000 Hz., with a peak around 2000 Hz. While the aural differences between centered and uncentered tones could be subtle for a novice performer, the visual differences are dramatic and unmistakable.

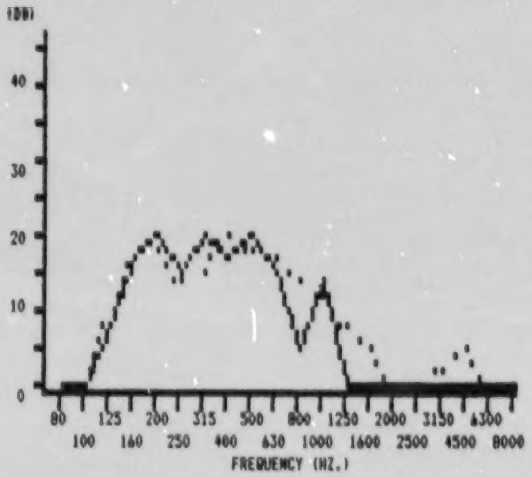
Figure 4. Spectrum graphs of two tones produced on a flute on the pitch F5 (698 Hz.). The tone produced by blowing into the tone hole is represented by the solid line, and the tone produced by blowing across the tone hole is represented by the dotted line.



Bassoon

A common problem with bassoon students is having too much of the reed in the mouth, resulting in an "unfocused" tone with extraneous noise. Figure 5 shows that a well produced tone was characterized by a strong fundamental frequency, a strong formant near 450 Hz., a weaker formant near 950 Hz., with a sharp decline in energy above 1100 Hz. This spectral pattern is generally characteristic of bassoon tones produced by professional players (Lehman, 1962). In contrast, the unfocused tone had less well defined peaks at the fundamental frequency and the formants. In addition the unfocused tone had some energy near 4500 Hz. which was not present in the well produced tone.

Figure 5. Spectrum graphs of two tones produced on a bassoon on the pitch G3 (196 Hz.). The solid line represents a well produced tone, while the dotted line represents a tone produced with too much of the reed in the player's mouth.



As Figure 5 illustrates, the spectral differences of the contrasting bassoon tones -- as shown by the spectrum graphs -- were not dramatic. This merely serves to point out that some types of timbre differences are not readily observed with a conventional graphic display. Typically, conventional spectrum displays show too much detail simultaneously. Other techniques of graphing the spectral differences might more clearly delineate the existing spectral differences. For example, spectral information might be more effectively displayed in some instances as pie graphs, seesaws, or color bars.

Discussion

Differences in tone quality between musical sounds can be made visible through a process involving real-time spectrum analysis and display via a microcomputer. Seeing the sound may aid students in perceiving aural details and in developing appropriate techniques for tone production. Some aural differences produce dramatic visual differences while others do not, although display formats are subject to alteration through software changes.

In actual applications of timbre visualization via microcomputer the teacher would provide the necessary physical directives to alter the tone

production. The process is inherently flexible, permitting its use by different teachers, who even may disagree concerning which tone qualities are desirable. The equipment is neutral concerning value judgments of timbre. The visualization technique could merely serve as an aid in teaching whatever concept of tone quality the individual teacher would choose.

The principle benefit of timbre visualization could lie in its making certain performance objectives more specific and measurable, thereby minimizing the use of confusing terminology. This, in turn, may help to establish pedagogy which is emperically based.

Goodwin, Allen. (1982, February). A pilot study for developing the singer's formant through use of an electronic biofeedback device. Paper presented at the meeting of the Music Educators National Conference, San Antonio, Texas.

Lehman, Paul. (1962). The harmonic structure of the tone of the bassoon. Unpublished doctoral dissertation, University of Michigan.

Roederer, Juan G. (1975). Introduction to the physics and psychophysics of music (2nd ed.). New York: Springer-Verlag.

Sundberg, Johann. (1977, March). The acoustics of the singing voice. Scientific American, 236(3), 82-91.

A PLATO-BASED ENGLISH FOR SCIENCE PROJECT

LANGUAGE INSTITUTE

DR. JANNIE J. BOTHA

ABSTRACT

In 1980 the Institute for Language Teaching at the University of Stellenbosch started on an English for Science project for Control Data.

Brief references will be made to the needs analyses, target population, and content of the courses. The main emphasis will fall on the one-on-one as well as group trial runs, which were conducted with members of the target population.

There will also be references to the results of both trial runs and especially to the ways in which data was collected during the different sessions.

A PLATO-BASED ENGLISH FOR SCIENCE PROJECT

1 INTRODUCTION

In 1981 at AILA in Sweden I gave a paper with the title "Computer-based Education and the teaching of English for Special Purposes: report of a project in progress". That paper was subsequently published in SYSTEM Vol. 10 No. 3, 1982.

In that paper I discussed the progress of a computer-based English for Science project because we had passed the halfway mark in the development work on the project.

Since the AILA Congress the project had been completed up to the group trial run stage. The purpose of the present paper, which is a sequel to my 1981 paper, is to provide information about

1.1 the design and development stages (an overview),

1.2 detailed information about the one-on-one and group trial runs,

1.3 the results of the trial runs but with the main emphasis on the results emerging from the group trials.

2 OVERVIEW OF THE DESIGN AND DEVELOPMENT STAGES

The design, and development of a CBE course have been one of the most exciting and rewarding experiences in my academic career. I can state this now when I look back at what has been done. There were many days, however, during the design and development stages when I felt like quitting!

The design phase of the project took approximately eight months. During this stage I performed a needs analysis which included the identification of the target population as well as determining the course content.

The target population that I identified was Zulu, Xhosa and Sesotho speaking students who had enrolled for a science course and who had only started their academic careers, but with inadequate English language skills. In other words, they were newcomers to the university. Furthermore they had to continue their educational activities in English as a second language.

The needs analysis revealed that the target population needed at least two English for Science courses, viz. reading comprehension, and academic writing. The needs analysis also showed the need for a listening comprehension course but the latter could not be accommodated within the time limits imposed on the project. It was therefore decided to continue with the development of a reading comprehension as well as an academic writing course.

The content of the courses was determined by analyzing the written work produced by members of the target population, by analyzing the textbooks that those students had to use, and by conducting interviews with faculty in the subject areas concerned.

The development of these courses took well over 20 months. One of the reasons for this was that my co-author was at a university some 600 miles away and all the manuscripts had to go to him as well. We further made use of a review team. The result was a delay in the progress but, we would like to believe, an improvement in the quality of lessons produced. The review team, by the way, consisted of language teaching specialists and in some cases programmers were also invited to participate in the discussions.

The programming of the courses were handled by our own programmers, trained at the Institute for Language Teaching. The programmers also had their fair share of frustration as far as the quality of the data lines were concerned. To be linked to a computer some 1000 miles away via telephone lines is no joke! Apart from this disadvantage the biggest advantage was to have programmers doing one's work just across the hall. That decreased tremendously the possibility for misunderstanding which could have resulted in work having to be reprogrammed.

3 THE TRIAL RUNS

In typical PLATO fashion we made a distinction between one-on-one, and group trial runs. After every trial run changes and adaptations were made to the courses. These changes involved design as well as programming. It must be pointed out, however, that major changes to the design of the courses will probably only be carried out after comprehensive summative evaluations have been done. These have not been attempted at this stage because the courses have not been implemented.

Because the development work was done in Stellenbosch for universities more than 600 miles away, one should have expected to meet some difficulties. The major problem was to find members of the target population at a specific point in time when a course was ready for a trial run. This was especially true in the case of the one-on-one trials. These had to take place during normal academic terms and we could therefore not expect students to travel long distances to take part in the trial runs.

The above-mentioned problem forced us to use students from our area who were in their final school year. We therefore

used Xhosa-speaking students in the afternoons after school. They were, by the way, delighted to have been introduced to PLATO!

The ONE-ON-ONE took place on the terminals in our Institute - risking the possibility of a communication breakdown! Some four students were invited to take part in the one-on-ones. Members of the review team helped to monitor the students by sitting in and advising them if they were stuck at a specific point in a lesson. The team members also noted such points by either using TERM-comments or making their own notes on paper.

The one-on-one trial runs were conducted immediately after a course had been completed. The advantage of this was that the work was still fresh in the minds of both the programmers and the authors. It was therefore easy to go back and make some changes in the lessons. The one-on-one merely served to indicate serious flaws in the flow and execution of the lessons.

At this point it is necessary to say something about the overall structure of the curriculum. The curriculum is PLM driven. This means that a student signs on in a PLM group. He then proceeds to take a test and from there he may select a learning resource which in this case is a CAI lesson. After he has completed a lesson he is returned to the PLM test. On completion of the test he is allowed to proceed to the next objective.

The GROUP TRIAL RUNS had to take place during university holidays because that was the only time when members from the target population were available. I have to acknowledge the assistance we received from the University of the Western Cape whose facilities we used for the group trials. We regarded it as of paramount importance to have access to terminals without being interrupted at any time during the trial sessions. Such an opportunity was only available during the university holidays. A terminal room was at our disposal for the entire period of the trial runs. That greatly facilitated the conducting of the trial runs.

We were fortunate to select members from the target population who were Xhosa-speaking. These four students were studying at the University of Fort Hare and they had the following "qualifications": they were enrolled in the Faculty of Natural Sciences; they were all first year students. Only one of them was studying computer science as a subject. It was also to our advantage that they all came from the Western Cape region, which is close to the University of the Western Cape, and Stellenbosch.

The main objective of the group trial runs was to obtain the maximum information about the students' reactions to PLATO, as well as to the lessons themselves.

Once again we assembled the review team, as well as the co-author. The team was responsible for helping the students where they could not continue in a lesson, and to write comments about grey areas in the lessons as well as the execution of the lessons. We were extremely fortunate to have had Dan la Berge from the U.S.A. with us. He spent 13 months in Stellenbosch as part of the development team. His knowledge and experience of PLATO programming rescued us from many unforeseen problems during the trial runs! It was inevitable that adjustments to the lessons had to be made during the trial runs.

Data about the trial run was collected through personal observations of the students by members of the team, as well as by means of a student data file attached to the lessons. This data file registered every keypress of every student and the information from it was very useful with regard to the execution of the lessons. It was also possible to make a preliminary item analysis of the test items used in the lessons and in PLM.

4 RESULTS OF TRIAL RUNS

After the group trial runs were completed it was back to the terminal to extract all the data stored in the student data file. This information was supplemented by the personal observations of the team members as well as the information obtained from informal interviews with the members of the target group who took part in the trial run.

It is only human to think that one's own work is without error! In the case of CBE lessons we definitely needed trial runs to confirm that perfect lessons are very seldom produced - let alone by newcomers to the field of CBE!

The results about the trial runs will deal with the following aspects:

* mechanical problems in the lessons
* editorial problems
* problems with prescriptions and directions
* areas of student confusion
* observations by the team members, as well as after discussions with students

4.1 MECHANICAL PROBLEMS

Generally speaking, problems of this nature are those built in by the programmers and which were not discovered during the busting of lessons - mainly due to familiarity with the lessons.

Some of the problems in this area were: final message at end of lesson was not applicable; suspected infinite loop because a student who could not answer a question could not continue with the lesson; end-of-lesson display was not available; completed learning resources were not marked as such, or learning resources were presented in an incorrect order.

4.2 EDITORIAL CHANGES

The main problem in this area was that references in the EST Study Guide and those on the screen did not coincide. This led to frustration among the students.

4.3 PROBLEMS WITH PRESCRIPTIONS AND DIRECTIONS

The system introduction to PLM (the starting point for students) was very difficult for members of the target group. It would be to the advantage of PLATO and CBE in South Africa, as well as the rest of South Africa, if this introduction could be freed from American jargon and rephrased in simple English!

4.4 AREAS OF STUDENT CONFUSION

The main area proved to be the problems the students had with the system prescription and directions referred to above. It was further discovered that a one-page introduction in the Study Guide could have saved the students' time.

4.5 OBSERVATIONS BY TEAM MEMBERS

It was found that the principle of Study Guides as complement to the on-line material seemed to work well. This principle was originally introduced because there was not enough room on the screen to present text and to discuss it.

It was amazing to see how often students made use of additional information which they could access through DATA, LAB, and HELP. This observation was also supported by the information in the student data file.

After approximately one hour at the terminal (having progressed to the second module of a course) the students showed increased confidence in their interaction with PLATO. This is significant if one bears in mind that all the students except one came from a non-computer background, or non-technological for that matter.

The above-mentioned results were phrased in general terms to facilitate the listening to and reading of this paper. Detailed information could be made available to interested people.

5 CONCLUSION

Although PLATO lesson design and development was a new field for me and my colleagues, it was an exciting and exhilarating experience. My sincere thanks to the company for trusting us to complete such a huge project!

DEVELOPING A CBI COURSE: THE PROCESS

Roland Garton, Mary Jac Reed,
George Reed, and Evelyn Stevens

University of Delaware

ABSTRACT

In 1981 and 1982, the University of Delaware's Office of Computer-Based Instruction (OCBI), under a contract to Control Data Corporation, helped produce computer-based instructional (CBI) materials for an entire first semester chemistry course. OCBI's successful team approach model for CBI lesson development was adapted and expanded to serve as a basis for OCBI's development process in LDEC Chemistry I. The resulting model included additional stages for overall course planning, conversion of lessons to a stand-alone system, and more carefully defined roles due to the size of the project and the number of groups involved. The purposes of this paper are to compare and contrast OCBI's process for single lesson development with the process of creating an integrated CBI course and to make recommendations regarding large-scale development models based on the experience. Careful records were kept with regard to staff and time requirements at all stages of the project. The results can help provide a foundation for planning similar projects in the future.

I. PROJECT BACKGROUND

From August 1981 to September 1982, the Office of Computer-Based Instruction (OCBI) at the University of Delaware, in cooperation with an editorial review board (ERB) of chemistry professors, and under contract to Control Data Corporation (CDC), designed and programmed computer-based instructional (CBI) materials for a first semester chemistry course. Well before development began, CDC perceived a need for CBI materials at beginning college level and outlined an entire curriculum of twenty courses for freshman engineering students, called the Lower Division Engineering Curriculum (LDEC), of which the Chemistry I course was a part. LDEC Chemistry I was comprised of forty-eight CBI lessons keyed to three standard freshman textbooks, to be delivered on the stand-alone Micro PLATO® system, with a written manual as a guide for the student using the materials. Early in 1981 the ERB, composed of five nationally-recognized, college level chemistry professors, met with CDC to make project decisions such as which topics should be presented on the computer and how long the lessons were to be. After that, OCBI worked with the ERB to design the lessons, and OCBI programmed them on the central system (for authoring ease) for conversion to the stand-alone system. OCBI was not significantly involved with the Fall 1982 pilot or the subsequent revisions, nor did OCBI have any direct input into the learner's guide that CDC developed. The scope of the project necessitated an involved development model, and for its part OCBI kept close track of both how well the model worked and how much time was spent in each stage of the process. The differences and similarities between the LDEC Chemistry I model and OCBI's standard model provide an interesting comparison, and the time study should help guide others involved in CBI development.

II. OCBI'S SYSTEMATIC APPROACH TO CBI DEVELOPMENT

The development of CBI materials shares much in common with the development process of other more traditional educational media. The stages of careful planning, development, review, evaluation, and dissemination apply to all endeavors of this type (Call-Himwich, 1977). In fact, some experienced CBI authors suggest that initial goals can be set with-

out considering the delivery mode (Nievergelt, 1980). In the development of CBI courseware, as in the development of books, filmstrips, and other educational materials, there is a need for instructional expertise, content expertise, and media expertise. In the more traditional educational areas, the author or authors often assume all three roles. In CBI development, however, authors can find themselves with at best limited experience of the medium. Assistance from a CBI specialist is usually required to adapt the content to the computer environment and to accomplish the programming.

At Delaware, OCBI's team approach has proven successful in developing and implementing CBI lessons (Seiler, 1981). The OCBI model is typical of standard CBI lesson development models, so it will be used to make a comparison with the LDEC Chemistry I course development model. The process is outlined below in reference to the chart in figure 1.

PLANNING

1. A faculty member identifies a need and, usually in consultation with OCBI, proposes a use for CBI in a course. The proposal includes a topic, a target audience, objectives, and an evaluation plan.
2. The faculty member supplies the basic outline of the content and the flow of the lesson(s).

DEVELOPMENT

3. The CBI expert on OCBI's staff adapts the content to best make use of the interactive computer environment, producing a page-by-page design of the displays and detailed lesson flow, referred to as a script.
4. After the faculty member, other faculty, and CBI specialists review the script, it is revised and given to a programmer. The programmed lesson is reviewed several times by the faculty member and CBI specialists. Revisions are made accordingly throughout the development process.

EVALUATION AND DISSEMINATION

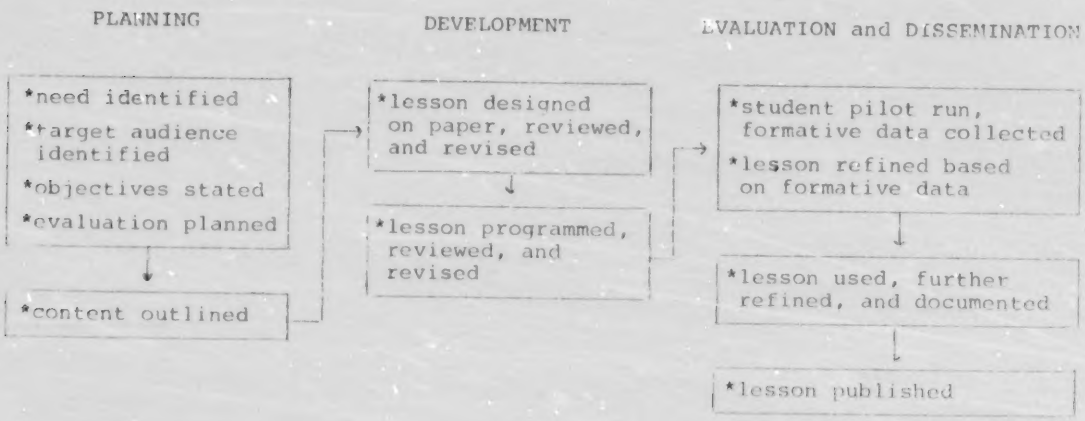
5. Formative data is collected during the initial student trial period. Based on this information, the lesson is further refined.
6. The lesson is usually used at the University while still undergoing the publication process. More reviews, revisions, and documentation occur at this stage. When it is approved, the lesson is published by a courseware company.

Roles overlap a great deal in practice, with each team member making content, design, and programming decisions, to varying degrees. Constant review by a wide variety of people and many revision cycles maintain the high quality of courseware produced by OCBI. Although development time varies greatly from project to project, a typical one- or two-lesson package is planned to span a year for the initial effort, with ongoing revisions based on evaluations and further insight.

III. LDEC CHEMISTRY I'S MODEL FOR CBI COURSE DEVELOPMENT

The LDEC Chemistry I development model was an experimental one based on standard models, with considerable expansion and

Figure 1: OCBI's Team Approach to CBI Development



formalization of the roles and the process. The stages are described below, referring to the chart in figure 2.

PLANNING

1. CDC identifies a need for a curriculum.
2. CDC plans overall scope of curriculum and identifies potential groups for development of curriculum components.
3. CDC meets with the ERB to determine objectives for the course, which objectives will be taught by PLATO lessons, length of each lesson, and instructional mode for each lesson.
4. Each ERB member writes a lesson content definition (LCD) for selected lessons describing more explicitly the content to be covered in each lesson and the instructional strategies to be used. Each LCD is reviewed closely by another ERB member, and the entire board gives approval to the set.

DEVELOPMENT

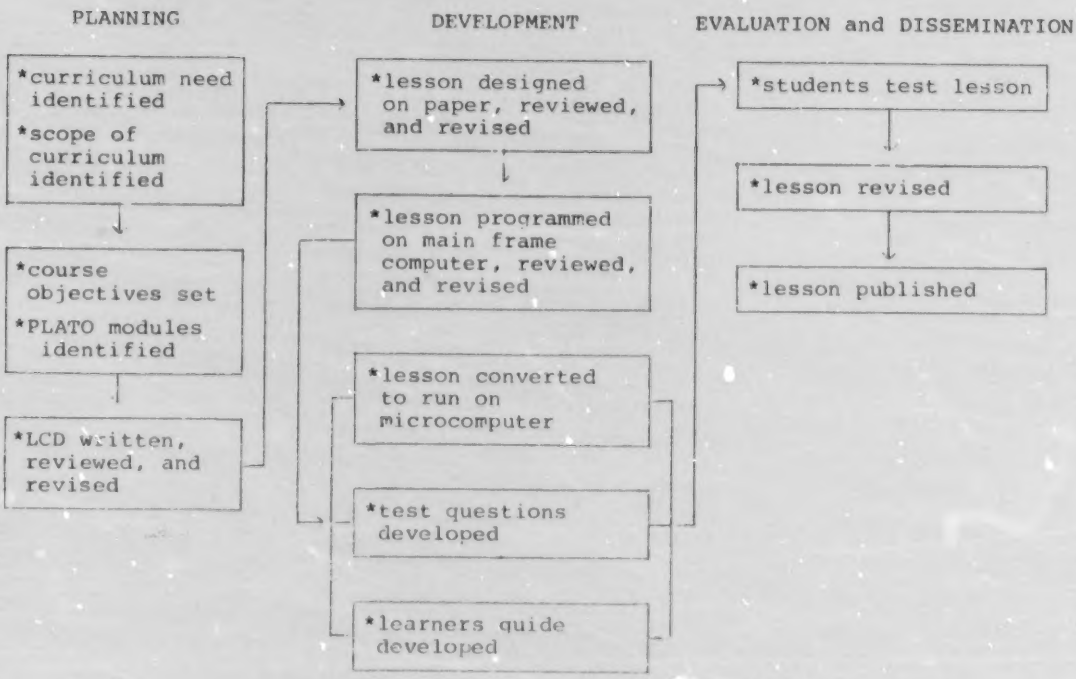
5. OCBI experts produce a script for each LCD which is reviewed and approved by the ERB with recommended changes.
6. OCBI programs the lesson to run on a central computer system delivering PLATO courseware. ERB members, with terminals at their schools, review, suggest changes, and approve the on-line lessons.
7. Simultaneously, CDC edits the on-line lesson and recommends changes.
8. OCBI revises the lesson and converts it to run on a stand-alone system supporting PLATO courseware.
9. Concurrently, CDC and the ERB develop questions for the on-line testing component.
10. Also concurrently, CDC develops a learner's guide for the course, with objectives for each lesson and suggested study approaches for the student.

EVALUATION and DISSEMINATION

11. Students at the five universities represented by the ERB go through the lessons completing evaluations and noting errors.
12. CDC checks for errors in each lesson's content and performance.
13. OCBI corrects errors found in the lessons.
14. CDC publishes and markets the materials.

Arriving at the development model itself took considerable time and effort. The expanded model worked differently than the OCBI model.

Figure 2: LDEC Chemistry I's CBI Development Model



COMPARISON OF THE TWO MODELS

A comparison of the two models outlined in figures 1 and 2 shows that although they follow the same overall structure, there are major differences between them. In the OCBI model, the need for CBI is first determined by a faculty member, with OCBI providing funding, coordinating the project, and setting guidelines for quality. In the LDEC Chemistry I model, all these roles were entirely or partially replaced by CDC as an external contractor. As a result, timeline and resource planning and adjusting became processes of sometimes lengthy negotiations, with the concerns of more parties to be dealt with. For OCBI, as part of a non-profit organization, the business and marketing concerns of such a venture proved a shift from its accustomed ways of working principally with educators. All groups involved became much more aware of timeline considerations, resource limitations, workable amounts of revisions, and communication needs among many geographically dispersed groups.

Where the OCBI model has faculty members on campus within easy walking distance for discussions about courseware, the LDEC Chemistry I model had five educators from across the country serving as content experts. While this increased the number of suggestions for each lesson and included the interests of different kinds of institutions, achieving consensus on issues among five diverse instructors without frequent discussions added an interesting dimension to the project. CDC's editing staff also reviewed and altered text in the lessons, adding another opinion to consider in the process.

Programming on a mainframe for conversion to a microcomputer-based system was a new process for OCBI. It did reduce authoring time compared to programming directly on the Micro PLATO system, even when conversion time is included. On-line lessons could also be more easily reviewed and commented upon by content experts, although some microcomputer graphics were not possible on the central system.

To point out important similarities, in both models faculty make content decisions. In addition, OCBI provides personnel for the roles of CBI specialists, both lesson designers and programmers, and numerous reviewers in both models.

The need for thorough planning, strong organization, and extensive communication in a project as large as LDEC Chemistry I is evident. There was a requirement for levels of overall planning at the curriculum and course levels that far exceeded that needed for development of one or a few lessons. Since the LDEC Chemistry I lesson users were to require minimal assistance, CDC planned and developed a student manual. Furthermore, in planning for vast amount of simultaneous input from several sources spread across great distances, the time devoted by everyone to writing comments at all levels and stages had to be taken into account. The sheer number of people, lessons, and comments in LDEC Chemistry I drastically increased the need for coordination and for time to reconcile and incorporate suggestions into the lessons.

IV. FINDINGS

For OCBI's part of the development process, accurate records were kept as to the time required for each stage, in terms of both number of hours and elapsed time. The results are summarized in figures 3 and 4. All times listed are averages; single lessons varied tremendously. On the average, it was found to take 350 hours for the OCBI team to script, program, and revise a typical half-hour lesson before students begin use, a development time to student time ratio of 700 to one. The total student contact time for the CBI lessons in the course, 1307 minutes, only slightly exceeded the predicted 1255 minutes, but individual lesson predictions were less accurate with the average lesson deviating from its projected lenght by 58.0%. The average development time per half hour lesson, 350 hours, exceeded original estimates by approximately 400%. Figure 3 lists only OCBI's part of the development process, which is broken down into the four stages of scripting, programming, conversion to the Micro PLATO system, and proofing. Revision, not listed as a separate step, occurs at all stages of the process.

Note that figure 3 shows total time spent but not elapsed time, an important distinction. For example, even though only eighty hours are required for the scripting process, factors such as time for creativity, review, revisions (including waiting for comments), push the elapsed time to six weeks. A list of average elapsed time for stages of the LDEC Chemistry I process is shown in figure 4. Estimating that it took five weeks for the ERB to organize and approve content, thirty-two weeks was the average time to ready a lesson for student pilot, excluding early planning steps. Including a semester-long pilot of sixteen weeks, and two weeks for subsequent corrections, the total average time per lesson was fifty weeks, nearly a year. However, the entire set of forty-eight lessons could not be started all at once, so the total development effort took

Figure 3: Average Time Spent on LDEC Chemistry I Development Stages

SCRIPTING stage:
- 40 hours for scriptor to script a lesson from an LCD
- 15 hours of coordination
- 20 person-hours for OCBI script reviews
- 20 hours for script revisions after OCBI review
- 10 hours for script revisions after ERB review

PROGRAMMING stage:
- 125 hours for programmer to program a lesson from a revised, detailed script
- 25 hours of coordination
- 15 person-hours for OCBI lesson review
- 10 hours for lesson revisions after OCBI review
- 20 hours for lesson revision after ERB review

CONVERSION stage:
- 20 hours for the converting team to convert a revised, approved lesson

PROOFING stage:
- 30 hours for proofing of the disk and testing of the lesson

TOTAL
- 350 hours/half hour CBI lesson

Figure 4: Average Elapsed Time on LDEC Chemistry I stages

LCD stage:
- 5 weeks for authors to write, review, revise, approve

SCRIPT stage:
- 6 weeks to write, review, revise
- 4 weeks for ERB approval of script
- 1 week to revise script based on ERB comments

PROGRAMMING stage:
- 6 weeks for full-time programmer
- 1 week for internal OCBI review
- 1 week for revisions based on OCBI review
- 2 weeks for review by ERB
- 2 weeks for revisions based on ERB review

CONVERSION stage:
- 2 weeks for conversion team to convert a lesson

PROOFING stage:
- 2 weeks for proofing and lesson testing

TOTAL
- 32 weeks/half hour CBI lesson

roughly a year and a half, including the pilot and subsequent revisions.

To accomplish the development, OCBI hired both full- and part-time lessons designers and programmers whose effort equalled that of fifteen full-time employees for OCBI's year of development. Part-time personnel played two important roles. First, experienced chemistry teachers in the areas served as part-time scriptors, closely supervised by OCBI's design experts. Second, students from the university were hired on a part-time basis to program the lessons.

V. RECOMMENDATIONS

The LDEC Chemistry I development model was a pioneering, experimental one, and a process of this scale requires a good development model, thorough planning, and extensive communication to keep all groups working well together. After being exhaustively involved with the process, the authors of this paper have a number of recommendations concerning each of these areas based on those steps that proved highly beneficial and on those where experience, in retrospect, might have suggested a different course.

A. Recommendations for Planning

1. Include contractor, content experts, design experts, and programmers at all planning stages to determine suitability of materials and approach to the medium as concerns content marketing.
2. Complete overall course planning before starting any development.
3. Include agreed-upon objectives, a testing strategy, design and editing conventions, and revision criteria in planning.
4. Define the target audience carefully. Wide, diverse target audiences increase the market for the courseware but present difficulties in defining teaching strategy.
5. Consult people who have experience with a similar development project.
6. Establish conventions for both scripting and programming in detail at the outset and adhere to them, to avoid retrofitted changes to lessons.
7. Allow for only a certain amount of change at each stage.
8. Establish a reasonable timeline that allows for foreseeable contingencies such as vacation and downtime, with target dates well ahead of absolute deadlines.
9. If lessons are to be presented in a definite order, develop them in that order to maintain consistency and to facilitate a logical flow.
10. Establish a measure for calculating length of lesson versus programming time, and have a good tracking mechanism to monitor lesson size and progress.
11. The testing strategy should measure student feedback, instructor feedback, and learning effectiveness.
12. Allow for a reasonable amount of revisions based on the student pilot.

B. Recommendations on the Development Model

1. Designate one person as the final arbiter on lesson content.
2. That person should supervise development and revision of the scripts as well.
3. Locate lesson designers and programmers at the site of the content expert.
4. The ERB should serve as a reviewing board rather than as a source of content.
5. Establish and agree upon the criteria for completion and acceptance of material at the beginning of the project.
6. Send one or two lessons through the entire procedure, including pilot, before major development, to test the model and adapt accordingly.

C. Recommendations on Communication

1. Clearly describe the development process and responsibilities of each party, and obtain their agreement before work begins.
2. Make all groups aware of the resources and limitations of all the other groups.
3. In the event of changes to the procedure or responsibilities, document changes for agreement by all groups.
4. Exploit the medium of the computer as available for rapid communication.

The LDEC Chemistry I project proved beneficial to OCBI in many ways, not the least of which was the experience gained in working on an innovative project in grand scale. The inevitable increase in demand for comprehensive CBI packages will dictate more large-scale CBI development efforts, and it is hoped that these recommendations will aid others involved in such efforts in the future.

BIBLIOGRAPHY

Caldwell, Robert M. "Instructional Design Consideration and Teaching Strategy Implementation Using Microcomputers." Monograph. Dallas, Texas: Southern Methodist University.

Call-Himwich, Eileen, and Steinberg, Esther R. Myth and Reality: Essential Decisions in Computer-Based Instructional Design (MTC Report No. 18). Urbana, Illinois: Computer-Based Education Research Laboratory (CERL), University of Illinois, [November 1977].

Feldhusen, John F. The Three-Stage Model of Course Design. Vol. 39 of The Instructional Design Library. Englewood Cliffs, New Jersey: Education Technology Publications, 1980.

Markle, Susan Meyer. Designs for Instructional Designers. Champaign, Illinois: Stipes Publishing Company, 1978.

Nievergelt, Jurg, "A Pragmatic Introduction to Courseware Design." IEEE, September 1980, pp. 7-20.

Seiler, Bonnie Anderson, "Applying a Systematic Approach to CBE Development in an Academic Environment." In 1981 Conference Proceedings ADCIS/CBI: Frontiers of Thought, pp. 305-310, Atlanta, Georgia: ADCIS, 2-5 March 1981.

PROGRAMMING A CBI COURSE: A CASE STUDY

Roland Garton, Mary Jac Reed,
George Reed, and Evelyn Stevens

University of Delaware

ABSTRACT

In the summer of 1981, Control Data Corporation decided to develop a series of Computer-Based Instructional (CBI) materials covering the first two years of a standard university engineering curriculum. Lessons were designed and programmed for the chemistry portion of the curriculum at the University of Delaware's Office of Computer-Based Instruction (OCBI) in conjunction with Control Data and an editorial review board. At OCBI records were kept on length of development time, skills of the programmers, and approaches taken in programming the lessons. This paper discusses team structures and programming practices that helped the process of courseware development. Further, the paper examines correlations among various CAI skills of the programmers and discusses why programming skill was not the greatest determinant of speed of development of the lessons.

The lessons were initially developed on the PLATO® network to be converted for delivery by floppy diskette on the Micro PLATO system. Conversion was accomplished in roughly four percent of original programming time. In a separate section this paper will discuss aids to the conversion process which apply more specifically to PLATO system to Micro PLATO system conversion.

I. PROJECT BACKGROUND

During 1981 and 1982 the University of Delaware's Office of Computer-Based Instruction (OCBI) took part in a project to produce computer-based instruction (CBI) on a very large scale. Control Data Corporation (CDC) desired to produce CBI for the diskette-based Micro PLATO system which, along with currently-used textbocks, would cover most of the courses that engineering students take in their first two years of undergraduate study. The overall project is referred to as the Lower Division Engineering Curriculum or LDEC project. The first three disciplines chosen for development were FORTRAN, physics, and chemistry, and OCBI was funded to develop the first semester of chemistry CBI materials, a total of forty-eight lessons.

To briefly outline the overall development model for LDEC Chemistry I, decisions about what topics to present on computers and what specific content to cover in each of those topics were made initially by CDC and an editorial review board. The content was then given to CBI specialists at OCBI for design and adaptation to the medium, after which the material was then programmed on a Control Data mainframe by OCBI programmers and readied for student use. After review and revision by the review board, CDC, and OCBI, the lessons were converted to run as stand-alone PLATO courseware. This oversimplification excludes numerous sub-steps and review cycles; a more complete treatment is offered in another paper (Garton, Reed, Reed, Stevens, 1984). This paper concentrates on just one part of the process--the programming of the lessons--which took place at OCBI.

II. THE PROGRAMMING PROCESS

A single programming process for LDEC Chemistry I would be impossible to describe neatly since a multiplicity of approaches was actually taken. In some cases a lesson was given to a single programmer. In others various routines were given to separate programmers to best employ their specific

talents such as graphics skills. In still others, lessons passed from one programmer to another as a result of factors such as changes in workforce. As the number of programmers increased so did the amount of time spent in coordination of their efforts.

Small teams of specialized programmers efficiently undertook some of the programming tasks that were similar for all lessons. For example, a large number of the lessons were converted to the floppy disk version by a few individuals who were familiar with the peculiarities of the Micro PLATO language. During review cycles, other programmers proved excellent at making minor changes to wording and graphics. A few quick programmers were given minimal long-range assignments to allow them to provide help in crucial situations, for instance, to insure completion of a lesson on time. This group was humorously called the "deadline-beaters."

Before the bulk of the programming was started a set of programming conventions and standards was developed, dealing principally with four areas: documentation, conversion, commands and structures to use, and commands and structures to avoid. These guidelines proved useful in clarifying standard coding practices initially and for aiding programmers who were added to the project over the course of time. The guidelines were also largely responsible for the ease of the conversion process. There was a lead programmer whose style served as a model for others, and who led the familiarizing of all the programmers with project procedures. He also served as a programming consultant.

Since each lesson design followed the same format with some identical pages, a skeleton lesson was created to serve as a template for all the lessons. A copy of the template to start each lesson provided the programmer with much of the structure in place and much of the initialization already complete. Furthermore, routines common to several lessons were stored in a single library that could be "borrowed" by any of the programmers, cutting down on duplication.

In order to make editing of the programs easier, several small utilities were designed. One of the most useful utilities, the "code-finder," allowed a programmer running a lesson in student mode to be brought directly to the code that generated a particular display, with a few simple keypresses. Another useful utility, the "brancher," allowed the programmer, while running a lesson, to be branched immediately to any other point in the lesson, without having to page through intermediate displays. Another option very useful for reviewers as well as programmers was the "skim" option, allowing users to bypass questions without responding. The standards, guidelines, shared routines, and overall team approach had a strong impact on the programming process in increasing standardization and decreasing individuality and possessiveness about code.

III. COMPARISON OF PROGRAMMING FACTORS

In trying to assess what factors among programmers had an effect on speed of lesson development, the authors of this paper (integrally involved with the project) rated each programmer according to 1) programming skill, 2) lesson design skill, 3) knowledge of chemistry, and 4) ability to work independently. The ratings were averaged to arrive at a profile for each programmer. Then twenty-seven of the forty-eight lessons were chosen randomly (by computer), and several factors for each were recorded: 1) the number of programmers responsible for at least 75% of the code, 2) the length of the program after being compiled, 3) the date the lesson was started, and 4) the amount of time elapsed between beginning and end of programming (here called "development time"). For lessons having more than one main programmer, the profiles of those programmers were averaged yielding a "programmer competency" profile for each lesson in the study. The "programmer competency" factors, when compared with the other aspects of the lessons, provide some results worth consideration.

Interestingly enough, there was very little correlation between length of program and development time ($r = 0.276$). The length of the programs themselves did not differ significantly [mean = 10,853 60-bit bytes (compiled code); standard deviation = 2,137]. Since the object was to determine what factors most affected development time, most of the comparisons are made using development time rather than length of program or length of program divided by development time, which would be programming rate. There also was not the correlation between programming skill and development time that one might expect ($r = 0.648$). This suggests that in a project of this size there are other factors besides programming skill that affect development time, factors such as length of time in a review cycle or nearness of a starting date to a deadline. The amount of shared routines certainly played a part in decreasing the differences in development time among the lessons, since large parts of some lessons could simply be plugged in as needed while other, perhaps smaller programs had to have programs and subroutines written from scratch.

Design skill of the programmer(s) was, however, significant as a predictor of development time; those lessons programmed by those with better design skills took less time in development ($r = 0.773$). Here it can be deduced that a programmer's design skill affects the design of the lesson enough to accelerate it through review cycles. Knowledge of chemistry had a similar effect on development time ($r = 0.777$), with the same explanation the most likely. As for starting date, those lessons started during the first five months of the project took far longer to complete than those begun later. With any project, the learning curve has its effect.

As the number of programmers on a given lesson increased, so did the length of time it took to develop the lesson ($r = 0.533$). This is a surprise since more person power should mean fewer months in development, and team approaches are geared toward using people's skills and talents efficiently. There are several possible explanations. It could be that the most highly-skilled programmers are so quick in all areas of programming that dividing up a program takes more time in collaboration than it does in actual programming. It could also be that the less experienced programmers were more likely to be doubled up on a lesson to keep it moving, leaving those lessons with slower programmers. Two or more programmers on a lesson might adversely affect the uniformity of the displays and increase the need for reviewers' comments, thus slowing the lesson in review cycles. A scatter plot of the number of programmers per lesson versus the development time is shown in figure 1.

To summarize the data, the overall "programmer competency factor" for each lesson was calculated by summing the design skill, programming skill, knowledge of chemistry, and independence-of-programmer ratings for the lesson. When plotted against development time as in figure 2, the correlation is evident ($r = -0.808$). Stated in short, programming skill in itself is not the greatest determinant of development time in a large-scale CAI development project; other skills make a greater difference in speeding progress.

IV. CONVERSION FROM MAINFRAME TO MICRO

The decision to program the lessons first on a mainframe that supports PLATO courseware and then to convert to the Micro PLATO system was based on the authoring advantages of the mainframe. The biggest advantage is in compilation time. On the central system a program can be edited, compiled, and running again in its new version within seconds, while the same process can take three to thirty minutes on the disk system. The central computer offers far more in the way of programmer and reviewer aids as well. The ability to automatically step through a lesson one command at a time does not exist on the microcomputer system used. On the central system, on-line consultants can monitor a programmer executing a lesson and can give immediate help. The "comments" option on the mainframe is indispensable for reviewers in that suggestions are automatically identified with the commenter's name, the lesson, and the display, and then the suggestions are stored for reference.

The lessons were intended to be delivered on a stand-alone microcomputer largely because of its advantages in costs. The Micro PLATO system has other advantages as well. Alternate font character sets can be loaded into a terminal from a disk in less than a second, a process which takes around twenty seconds on line. In the micro lessons several sets of alternate characters could easily be swapped in and out of memory. The speed of graphics is greatly enhanced in the micro version, allowing three-dimensional molecular rotations not possible on the central system. At the time of on-line review, however, many advantages of the micro system became disadvantages for the reviewers since effects were not reproducible on the mainframe.

The central system has two main aids for conversion to a Micro PLATO delivery environment. The first is a Micro PLATO language emulator on the central executer, whose purpose is to execute micro code on the central system. The emulator only became available to the project after a large amount of work had progressed. It did not quite offer the full range of the micro language commands, and it lacked some debugging aids, so it remained essentially unused. The second aid is an automatic conversion program. When code is written in the mainframe language especially for the conversion program, it can convert a full 95% of the code, leaving only the most complicated routines for calculating and database manipulation unconverted. As an additional aid, OCBI developed a set of guidelines for programming with an eye toward automated conversion. Early in the programming process a few sample lessons were converted and the guidelines were improved. An abbreviated version of the guidelines appears in figure 3. While some of the guidelines are quite specific to the languages involved, others apply to any code written for conversion to another language. In addition, the conversion process itself was detailed in another set of guidelines.

As a result of system advantages and planning the average lesson took about 15 hours for the actual conversion, or 4% of

the original programming time, and 95% of the code was converted (or left unchanged) automatically. Once converted, however, each lesson had yet to undergo more review/revision cycles.

As a comparison, in 1980 an initial attempt to convert sophisticated PLATO courseware to the same micro environment was undertaken. The lessons converted in 1980 were more complex in data structure than the LDEC Chemistry I lessons, and that effort was begun without several advantages. In 1980 the micro language was less like the mainframe language, requiring more code to be rewritten. By 1981 a fairly reliable converter lesson had been developed. In between the two projects a wealth of cumulative experience allowed faster solutions to problems and a set of guidelines to ease conversion. Without these advantages, the 1980 lessons took half as long to convert as they did to program originally, and only about 70% of the original code could be easily converted (Garton, 1980).

The LDEC Chemistry I project provided many opportunities to experiment with production models and procedures. It is hoped that the insight gained will continue to be useful to OCBI and to others in the field. The project also provided OCBI the opportunity to increase the number of skilled programmers that provide a backbone for future development. Developers at OCBI look forward to applying what was learned with LDEC Chemistry I to other large-scale development projects, as CBI becomes more consistently integrated into a standard college curriculum.

Figure 1: Programmers per Lesson vs. Development Time

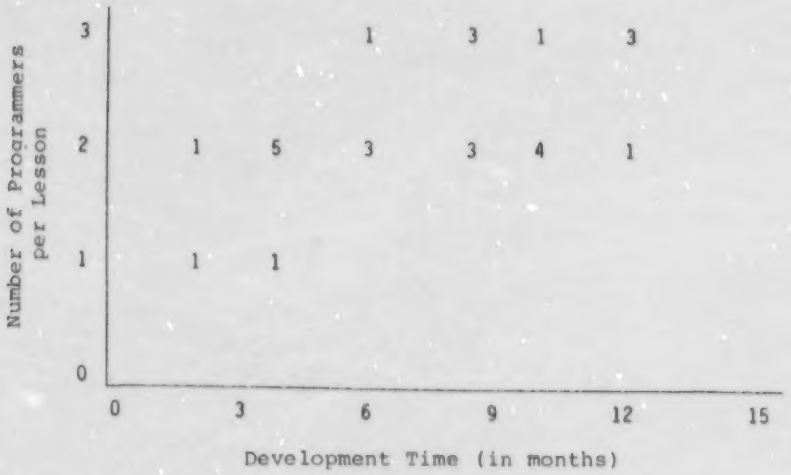


Figure 2: Overall Programmer Competency vs. Development Time

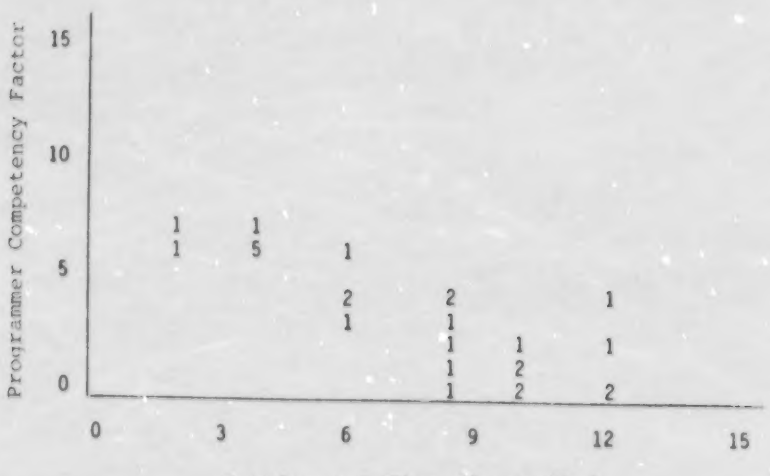


Figure 3: Programming Mainframe PLATO Courseware for Micro PLATO Conversion

I. Displaying Text with -write- and -writec-

- Embedded -show-s can be used.
- Avoid long -writec-s.

II. Screen Locations, -draw- and -at-

- Use wherex and wherey rather than where.
- Do not use wherex and wherey in a -draw- command, rather, store them in variables first, and use the variables.

III. Displaying Values with -show-

- Use -show- only for integers.
- Use -showt- for all fractional values.
- Use -showa- only in the two tag form.

IV. Sized and Rotated Displays

- -size- & -rotate- may be used with relative commands.
- Do not use sized or rotated text (except as below).
- Text (size 1) may be rotated 90°.
- Mode bold can be used if acceptable by lesson design.

V. Using -calc- and Random Numbers

- Use ** rather than super for exponentiation.
- Do not use TUTOR array operations.
- Do not use -calc- to assign a character string to a variable; rather, use -pack-.
- Use only the two tag form of -randu-, -setperm-, -randp-.

VI. Judging

- Be sure to enclose synonyms in parentheses.
- Do not use -touch- or -touchw-; for touch use -keytype-.
- Use -specs okcap- rather than -bumpshift-.
- Use -putd- rather than -put-.
- Note where student responses longer than three words are anticipated.

VII. Branching and Sequencing

- Use only integer variables with -doto-.
- Always have a -next- command in each main unit.
- Do not pass "charstrngs" as arguments.

VIII. Defining and Documenting Variables

- List the maximum value each variable will hold.
- When using segments:
  - The byte size should be 16 or 8.
  - Explicitly document the total number of segments required.
- Use segments, rather than bits whenever possible, but when bits are necessary, give a full description of bitsetting routines.
- Use floating point variables when working with numbers greater than 32768 or less than -32769.
- Make only one definition per line.

IX. Some Things to Avoid

Avoid:
- complex answer judging.
- using -specs- as a marker.
- using -store- or -storen-.
- fancy routines when a simple one will do the same thing.
- complex calculations and the use of trig functions when possible.
- very small -do-ne units, and large, multipurpose -do-ne units.
- very long units.
- branching across very large sections of code.
- tricks and "cute" routines.
- common and storage variables.
- bitsetting routines, and bit manipulations.
- -comput-, it does not act the same in #mTUTOR.

X. General

- Include a -charlim- command set to the number of the highest character slot in use. This is a #mTUTOR command so '*' to comment it in the TUTOR version.
- Refer to mpxlator for a list of commands that it can handle. Stick to these commands whenever possible.

XI. Documenting

- Identify code that does not follow the guidelines or that needs to be checked carefully during the conversion process with a "%" in a comment statement. This will serve as a flag for things to check during conversion.

BIBLIOGRAPHY

Garton, R.; Reed, G.; Reed, M. J.; and Stevens, E. "Developing a CBI Course: The Process." To be published in the 1984 ADCIS Conference Proceedings, Columbia, Ohio: ADCIS, 14-18 May 1984.

Garton, R. and Silver, J. "Approaches to Converting PLATO® Courseware to Micro-TUTOR: A Case Study." In 1981 Conference Proceedings ADCIS/CBI: Frontiers of Thought, pp. 162-166. Atlanta, Georgia: ADCIS, 2-5 March 1981.

A Model for the PLATO® Services Organization in a Large, Multi-System CBE Environment

Rae D. Stabosz
Jessica Weissman

University of Delaware

ABSTRACT

The PLATO Services Organization (PSO) of the University of Delaware's Office of Computer-Based Instruction (OCBI) has developed a model for consulting which concentrates on author training and user services as its primary functions. This model has been used successfully while the PLATO network was the major development system for CBE at Delaware and continues to serve as the PLATO network takes its place in an expanding CBE environment which utilizes other mini- and microcomputer systems for courseware development. This paper summarizes the model for PSO in this large, multi-system CBE environment. Included is a list of PSO competencies which delineate the scope of expertise and information dissemination expected within the PLATO Services Organization under this model.

I. INTRODUCTION

A fourfold function has been identified for the PLATO Services Organization under the University of Delaware's model for author training and user services: consulting, author training, centralized information resource, and publication review. Under the model, all four functions have as their primary goal the enhancement of learning for the author and user. This means that while carrying out these four functions the PLATO services consultant will seek not only to meet the immediate need of the author or user but to do so in a manner that fosters growth in knowledge of the PLATO network and the TUTOR language. Rather than concentrating on the "quick fix," the consultant will, when appropriate, stimulate investigation of the problem at hand by the author or user, so that the solution to the problem will occur with the active involvement of the consultee rather than with his or her passive reception of the solution. There are times in which passive reception of information is the most appropriate response to a given problem, and some of the four PSO functions lend themselves more realistically to passive than to active involvement of the author or user. In general, however, the model insists on the desirability, whenever possible and appropriate, of active involvement of the consultee in the case for which a consultant is called.

II. CONSULTING

PLATO services consultants deliver on- and off-line consulting services from 8:00 am to 8:00 pm Monday through Friday, for a total of 60 hours of on-duty consulting time per week. Off-line consulting can be scheduled for those wishing in-person help with problems, while on-line consulting is encouraged for the kind of immediate problems that occur as a user is working on the system.

The University of Delaware has as its consulting clients both customers from outside industry and student and faculty programmers and designers from within the University. Its model for consulting encourages consultants to know their clients and be ready to offer an individualized style of consulting, paying special attention to the new user. "Know thy consultee" might be a watchword for on-line consulting. Many users have great difficulties in communicating their problems over the terminal. This is particularly true of new authors and instructors. It takes a certain amount of time for a new user to get comfortable using an unfamiliar keyset; in addition, a new author might be a poor or slow typist. Both these considerations, coupled with the

sequences of keypresses that must be learned in order to use the system software for on-line consulting, can make a new user shy about displaying his or her "clumsiness" under the eyes of the invisible consultant. Patience and friendliness on the part of the consultant can make a difference in the initial experience of the new user.

Considerations of new users aside, there are definite consult strategies offered by the Delaware model. Often the consultee does not know what his or her problem is and simply finding the problem is the main task of the consultant. Here, using standard debugging techniques to isolate the error works the best. Step mode, short test units, and displaying variables are often overlooked by young programmers. Consultants are advised to check for the obvious first. It can't always be assumed that consultees have checked out simple solutions, and even if they have, they may have overlooked an obvious solution.

Many times a consultee will ask how to set up a program. Some consultees learn well by example and others learn better by being led through the process of setting up the code. Checking to make sure that he or she is being understood while explaining things is important in determining the approach to be used by the consultant.

Consulting is an opportunity to teach high standards in design and programming. Consultees usually appreciate comments that can better their lesson even though they may not have asked a question specifically about it. In this sense, consulting can be seen as an extension of the seminars.

III. AUTHOR TRAINING

Author training is a very important function of the University of Delaware's PLATO Services Organization. Each semester, a series of seminars is offered at no charge to university faculty, staff and students. In addition, student programmers at OCBI are expected to improve their skills and update their knowledge by taking one of these seminars per semester.

The PSO staff teaches only those seminars which are related directly to CBE programming on the PLATO network. Included among these are an introduction to TUTOR programming, intermediate level TUTOR programming, PLATO Learning Management (PLM), microTUTOR, TUTOR programming review, programming problems seminar and special topics in TUTOR. Other seminars on PLATO lesson design are offered by design specialists, making a full range of seminars for CBE on a PLATO network.

The seminars in TUTOR and microTUTOR programming are offered as the normal way for interested university faculty, staff and students to acquire an author signon and begin a directed path toward proficiency in TUTOR. OCBI employs over one hundred student programmers, and many of these program exclusively in TUTOR. Although there does exist an option for students to learn TUTOR in a directed course of self-study, the normal route for learning TUTOR is through the seminars. In this way, the consulting staff is responsible for the bulk of the training of student programmers, and the PSO staff is able to give input on the skills and potential of student programmers to OCBI project coordinators seeking to hire part-time programming staff.

TUTOR programming reviews are critiques of the TUTOR code of lessons currently being programmed by university development projects. A section of a lesson is identified for code review and prints are distributed a week in advance to anyone wishing to attend the review. Under the guidance of a consultant, the code is then reviewed intensively during a 2-hour session by all participants, and its efficiency, documentation and maintainability is discussed. The sessions are meant to function as a means of communication between programmers about programming itself, as well as a learning experience. The audience is typically a mixed group of experienced programmers and novices, and the consultant must direct the session so that the talk does not become so esoteric that novices are left far behind nor so basic that the experienced programmers are bored.

Programming problems seminar is another approach to getting programmers of all levels together to talk about the programming process. A week in advance of a session, a problem is announced publicly in one of the PLATO notesfiles. Participants are encouraged to work on a solution to the problem over the week's time, and to bring copies of the solution to the session. At the session, participants exchange solutions and discuss the merits and disadvantages of the various approaches taken in solving the problem. In this way, programmers can see first-hand how other programmers might vary in approaching a solution to a given problem, and programmers are able to "show off" their favorite techniques and styles of programming. The role of the consultant is to coordinate the discussion, the logistics of which will change according to how many participants have prepared a solution. Usually, the consultant and between two and five programmers come into the session with a

prepared solution, and all participants take home copies of all solutions.

Special topics in TUTOR are two-hour sessions given by OCBI staff members which focus on some small aspect of programming on the PLATO network. A presentation and demonstration of the feature is given, interspersed with and followed by discussion. These special topics sessions are intended to fill in the gaps left by time restrictions in formal TUTOR programming seminars, to expand in depth on topics covered in those seminars, and to allow OCBI staff members to demonstrate skills developed while concentrating on special areas of programming. The role of the consultant in these seminars is usually one of advisor. Some sessions are given by the PSO staff, but more typically a consultant will solicit the interest of another staff member in giving a presentation in his or her given area and will then work with the presenter in planning the scope and organization of the session. This is especially true where the presenter has not had experience in teaching. For those staff members who have taught as part of their regular duties, the consultant functions mainly in a coordinating capacity, organizing the scheduling of the session and making sure of room availability, etc.

IV. CENTRALIZED RESOURCE

By custom and practice the PLATO Services Organization on the Delaware PLATO network has served as a centralized source of information and reference. Secretaries routinely direct information-seeking telephone calls and visitors with questions to the consultant on duty when the staff member who ought to take the call is unavailable. Consultants usually make themselves available in the centrally-located consulting office during their normally scheduled on-duty consult hours. This centralized location and the availability of a consultant from 8:00 a.m. - 8:00 p.m. each working day facilitates use of the PLATO Services Organization as a centralized reference source.

During the last three years, Delaware's Office of Computer-Based Instruction has been extending its CBE development efforts to areas outside of the PLATO network. Major development work is being done on a VAX series 11/780 super-mini computer, on the IBM Personal Computer, and on the Apple II+ and IIe. The PSO function of centralized resource has extended in a very limited basis to these other development areas. Direct consulting is not done by PSO consultants for programmers working in non-PLATO environments, although some PSO consultants, whose professional duties have not been 100% devoted to PSO, have been assigned teaching and workshop-planning roles in the micro development area. A large microcomputer laboratory and adjoining staff room serves as the centralized information resource for micro development projects and author training for microcomputers. Still, during the transitional phase, when a parallel organization of reference people and seminar offerings is being established, patterned largely after the PSO model, PLATO services continues to receive and refer inquiries in the micro development area.

As a centralized resource, PSO does not solicit active involvement of its clients in the services it provides. Unlike consulting and author training, as well as publication review to some extent, the function of centralized resource normally calls for active dissemination of information on the part of the consultant and passive reception of same on the part of the client. Consultants can, where appropriate, suggest additional sources of information once the immediate question has been answered, but active consultee involvement is not solicited.

V. PUBLICATION REVIEW

The final function of the PLATO Services Organization is publication review. Until recently, courseware developed at the University of Delaware was prepared for publication according to Control Data Corporation's "Technical and Mechanical Standards for Computer-Based Courseware" (Control Data Corp., 1982) and submitted to Control Data for publication. In the spring of 1983, however, the Office of Computer-Based Instruction chose to terminate its publishing agreement with Control Data in favor of acting as its own publishing agent. Using existing CDC standards as a model, OCBI then developed a set of publication standards and its own step-by-step process of final review of lessons for publication. This step-by-step process is the final phase of the regular process of ongoing review of lesson scripts, design, screen displays and code followed by development projects using the standard system's approach to courseware development (Seiler, 1981) employed at the University of Delaware.

The role of the PLATO Services Organization in the publication process is that of review of the code of the completed lesson according to the publication standards developed at Delaware. These standards are essentially the same as those found in Control Data's document of

publication standards, mentioned above, with the following OCBI additions:

1. Special resources: if the lesson is not going to be restricted to intelligent terminals, the lesson must not use the -fill-, -beep-, and -mode inverse- commands unless "zttype" is used to check and provide alternate code for non-intelligent terminals.

2. Character sets: if "zreturn" indicates that the charset load has failed, the lesson should try to load again, but not an infinite number of times (although an infinite number of times is all right when STOP was pressed). If it fails after a given number of times, give a message and end the lesson.

3. Condense errors: in files which are partialled out because they are not designed to condense when used alone, one condensable block should explain what the proper entry lesson's file name is and should provide -jumpout- to it.

4. -term-s: the TERM key must work at every pause.

5. Variable file references: although the lesson may not refer to any other file by placing the name of the file in a variable, the name of a file may be a defined constant.

6. Documentation:
   a. The purpose of each variable should be described with its definition, in the define set.
   b. For main units, a text block showing the unit flow will serve instead of stating the purpose of the units.
   c. The above requirements are waived if the current programmer in charge is no more familiar with the lesson than anyone else would be.

VI. PSO COMPETENCIES

The PLATO Services Organization has outlined a set of competencies necessary for carrying out its fourfold function (see Appendix.) Eventually the outline will be fleshed out and criteria for judging whether the competencies are being met will be developed. For now, the outline is useful as a basis for knowing the scope of services provided by PSO. While it is desirable that each consultant be as skilled as possible in each area, it has been efficient to develop complementarity and specialization of consultants in the various areas.

APPENDIX: OUTLINE OF PSO COMPETENCIES

I. Knowledge of the PLATO Network

   A. Acount management

   B. Site management

      1. System site lessons

      2. uofdel site lessons

      3. Notesfiles for communication about site management

   C. OCBI policies

   D. OCBI prices

      1. All exact price quotes referred to services

2. Give out approximate costs of:
   a. purchases terminal
   b. rented terminal
   c. port cost, per year
   d. phone line connection

E. Editors
   1. TUTOR editor
   2. Display editor (ID/SD)
   3. Notes editor
   4. Database editors (common, nameset, dataset)

F. TUTOR programming language
   1. Text display
   2. Sequencing
   3. Judging
   4. Calculating
   5. Iteration
   6. Variable definition; scope of local and global vars
   7. Defined functions
   8. Graphing
   9. List manipulation
   10. Database structuring and manipulation
   11. Documentation and maintainability

G. MicroTUTOR
   1. Differences and similarities to TUTOR
   2. Techniques for converting courseware from TUTOR to microTUTOR (Garton & Silver, 1981)

H. PLATO Learning Management (PLM)

I. Communications
   1. Notesfiles and notesfile sequencer
   2. Communications -term-s: term-talk, term-consult, term-confer
   3. Personal notes
   4. Lessons teaching communications

J. Student Groups

1. mrouter
2. plmrouter
3. the University of Delaware catalog routing system

K. GUIDE

L. Documentor

M. University of Delaware quiz drivers

N. "notesedit"

O. Printing

II. Lesson Design

A. Design principles outlined in "Guidelines for Creating PLATO Lessons" (Seiler, Note 1)

B. Knowledge of OCBI design resources

1. internal design specialists
2. professional journals and reference books
3. software on design

III. Consulting

A. System consult software: lesson "c", "constats"

B. OCBI consult software: lesson "consults"

C. Consulting techniques (see section II, above)

D. OCBI consult standards: acceptable response time, acceptable "calls missed" limits

IV. Author Training

A. Teaching skills

B. Teaching experience

C. Author tracking system

D. Teaching materials and resources

1. Beginning TUTOR Programming - Williams (Note 2)
2. "Tougher Topics in TUTOR" - Mulford (Note 3)
3. Fundamentals of PLATO® Programming - Davis
4. The TUTOR Language - Sherwood

V. Centralized Information Resource

A. Who knows what: services personnel, site managers, project managers, coordinators, design people, microlab staff, editors

B. Lessons concerning staff responsibilities

C. Lessons concerning project and faculty developers

D. Knowing when to refer and when to respond directly

VI. Publication review

A. OCBI publication process

B. Notesfiles and lessons on publication process

C. OCBI publication code standards

REFERENCES

1. Seiler, B. Guidelines for Designing PLATO Lessons. Available at the Office of Computer-Based Instruction, University of Delaware.

2. Williams, D. Beginning TUTOR Programming. Available at the Office of Computer-Based Instruction, University of Delaware.

3. Mulford, G. Tougher Topics in TUTOR. Available at the Office of Computer-Based Instruction, University of Delaware.

BIBLIOGRAPHY

Control Data Corporation. Technical and Mechanical Standards for Computer-Based Courseware. Control Data Corporation, 1982.

Davis, C. Fundamentals of PLATO® Programming. Urbana, IL: University of Illinois, 1980.

Garton, R. and Silver, J. "Approaches to Converting PLATO® Courseware to Micro-TUTOR: A Case Study." In 1981 Conference Proceedings ADCIS/CBI: Frontiers of Thought, pp. 162-166. Atlanta, Georgia: ADCIS, 2-5 March 1981.

Seiler, B. A. "Applying a Systematic Approach to CBE Development in an Academic Environment." In 1981 Conference Proceedings ADCIS/CBI: Frontiers of Thought, pp. 305-310. Atlanta, GA, 2-5 March 1981.

Sherwood, B. The TUTOR Language. Urbana, Illinois: Bruce Arne Sherwood, Computer-Based Education Research Laboratory (CERL), University of Illinois, 1980.

LEARNING ABOUT SUPERCOMPUTERS ON A MICROCOMPUTER WITH NO KEYBOARD: A SCIENCE MUSEUM EXHIBIT

Mary L. Stoddard and Bill L. Buzbee

Los Alamos National Laboratory

ABSTRACT

A microcomputer exhibit was developed to acquaint visitors of the Los Alamos National Laboratory's Bradbury Science Museum with supercomputers and computer-graphics applications. The exhibit is highly interactive, yet the visitor uses only the touch panel of the CD 110 microcomputer. The museum environment presented many constraints to the development team, yet the five minute exhibit has been extremely popular with visitors. Design details of how each constraint was dealt with to produce a motivating and instructional exhibit are provided. Although the program itself deals with a subject area primarily applicable to Los Alamos, the design features are transferable to other courseware where motivational and learning aspects are of equal importance.

Introduction

In 1982-83, the Bradbury Science Hall and Museum at Los Alamos National Laboratory was redesigned to be self-guiding and interactive. One of the new exhibits was a microVLATO station. The primary objective of this exhibit was to tell the museum visitor about the critical role that computing technology, especially high-performance computers, plays in scientific research. The constraints of the museum environment presented uncommon challenges to the exhibit-development team.

Recent evaluation studies of exhibits in science museums have indicated the need for a feedback loop between the visitor and the exhibit in order for the exhibit to be instructionally effective (Kimche, 1973; Luetchford, 1976). Visitors in many museums are now taking an active role in seeking new information (Miles, 1978). In the area of computing, some museums are using microcomputers to help children learn about programming (e.g., Patterson, 1983).

This paper begins with a description of the goals and constraints of the project and then presents the exhibit specifications, the development process, and a description of the program. The paper concludes with a discussion of the visitor reaction to the exhibit and provides a summary of the project highlights. This project may be of interest to all courseware developers regardless of environment, as some of the design features of the program may prove effective in learning-center environments as well.

Goals and Constraints of the Project

A hand's-on exhibit was needed in the new Bradbury Science Museum to teach visitors about the role computers (especially supercomputers) play at Los Alamos National Laboratory. The Los Alamos Computing Network is one of the largest in the world; it currently contains several supercomputers and scores of smaller computers to support the intensive research demands of the Laboratory's scientists. Sophisticated computer graphics are used in much of the mathematical modeling performed on the Network computers.

The intent was to have the exhibit be: highly interactive, short (maximum of five minutes to complete), graphical, fun, and instructional. Kimche (1973) claims that "effective teaching exhibits should integrate both learning and motivational aspects." Also, the equipment could not require special power and cooling facilities. Although a growing segment of the population is familiar with computing technology, the decision was made to seek a user interface for the exhibit that older people might find easy to use, even unintimidating. This objective ultimately led to the selection of a touch-sensitive

screen for interactivity. Another important reason for this decision was that because of people's curiosity and tendency to experiment, the equipment had to be housed in such a fashion that relatively few buttons, dials, etc. would be available for manipulation. And to help the visitor recall the exhibit, it was desirable to provide some sort of personalized souvenir to take home.

The exhibit needed to attract and hold the interest of visitors ranging from school children to scientists. In addition, many of the museum visitors speak Spanish and thus ideally the exhibit would present information both in English and Spanish. Because the museum is designed to be self-guiding, the exhibit had to be self-sufficient and require low maintenance. The museum is small, yet often large groups of visitors tour at one time. The exhibit therefore needed to accommodate from one to many visitors. Also, the personnel who operate the museum are typically not computer professionals. Thus startup and shutdown procedures of the system had to be simple.

Hardware and Software Specifications

A committee of computer scientists met with museum designers and determined that a microcomputer with graphics capabilities would best meet the goals and constraints of the exhibit. The microPLATO, CD 110 system, was selected and the microTUTOR language was identified for authoring. The exhibit would contain two CD 110s, and a printer shared by the two microcomputers. Software for each CD 110 would be identical.

The CD 110 has a touch panel and it was determined that the keyboard not be used by the museum visitors; all interactions would be done by touch. On the sides of the exhibit structure would be displayed examples of the high-resolution-color computer graphics generated at the Laboratory, as the CD 110 does not have color graphics capabilities.

The software specifications were:

- use of the touch panel for every interaction;
- automatic signon to exhibit;
- ability to move back, forward, or get out on every frame;
- automatic return to the beginning of the program if station is idle for two minutes;
- personalized souvenir, with visitor's first name and today's date, requested by touching the screen (not the printer); and
- option of whether English or Spanish version is desired.

The Development Process

A committee of seven computer scientists and instructional designers was brought together to design the program. A programmer was identified to do the programming. And a translator was selected to do the Spanish translation. The committee met and determined that the program would have two major objectives and consist of two main parts: (1) Monte Carlo Simulation: The visitor will be able to describe why the Laboratory needs superfast computers, and (2) Computer Graphics: The visitor will be able to list several computer graphics applications at Los Alamos. The committee then divided into two development subcommittees, with the instructional designers coordinating the effort and communicating with the programmer.

Each subcommittee wrote objectives and devised screen displays for their part. After three iterations, the hard copy screen designs were turned over to the programmer. The program underwent formative evaluation, and then the Spanish translation was done and implemented.

The Program

The program consists of 56 frames, many of which represent progressive disclosure through pauses. The visitor is welcomed to the exhibit (see Figure 1), and then is provided two frames of instructions on use of the touch panel. A touch menu is then presented providing the visitor with the option of learning about graphics or superfast computers first.

Figure 1
Welcome frame.

The graphics portion begins with a quick animation and consideration of the question, "How are computer graphics used to understand the 200,000 pages of data produced at the Laboratory each day?" The visitor is then given a problem: Determine which is easier to interpret, a table or a graph presenting the same data (see Figure 2). If the users touch the box labeled "graph," they proceed to explanations of color graphics on the side panels. If the "table" box is touched, the visitor branches off to an explanation of why the graph is easier to understand. The graphics portion concludes with the visitor touching letters on the screen to spell his/her name and then requesting a souvenir printout (see Figure 3).

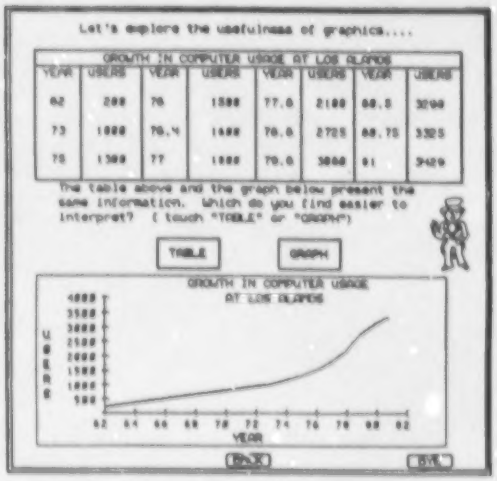
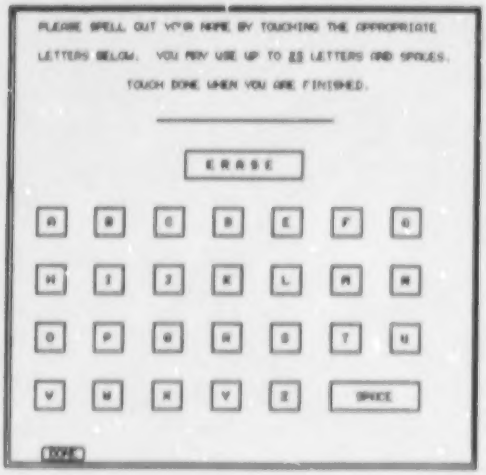

Let's explore the usefulness of graphics....
GROWTH IN COMPUTER USAGE AT LOS ALAMOS
YEAR USERS YEAR USERS YEAR USERS YEAR USERS
62 200 70 1500 77.0 2100 80.5 3200
73 1000 75.4 1600 78.0 2725 80.75 3325
75 1300 77 1800 79.0 3050 81 3429
The table above and the graph below present the same information. Which do you find easier to interpret? (touch "TABLE" or "GRAPH")
TABLE
GRAPH
GROWTH IN COMPUTER USAGE AT LOS ALAMOS
USERS
YEAR
62 64 66 68 70 72 74 76 78 80 82

Figure 2.
Why graphics are easier to interpret than tables.

The superfast computers portion begins with consideration of the question, "How can rather simple computations require tremendous amounts of arithmetic?" Sequentially, the speed of a supercomputer vs. a microcomputer vs. a hand held calculator are compared (see Figure 4). The computational method called Monte Carlo is then introduced to show how some complex problems are solved using superfast computers. The Monte Carlo method is used to compute the value of π through a sequence of darts thrown randomly at a dart board. The visitor may select the number of darts thrown, and sees the precision of the calculation of π increase with the number of darts thrown (and hence the number of arithmetic operations performed) (see Figure 5). The exhibit concludes with a brief discussion of the


PLEASE SPELL OUT YOUR NAME BY TOUCHING THE APPROPRIATE LETTERS BELOW. YOU MAY USE UP TO 25 LETTERS AND SPACES. TOUCH DONE WHEN YOU ARE FINISHED.
ERASE
A B C D E F G
H I J K L M N
O P Q R S T U
V W X Y Z SPACE
DONE

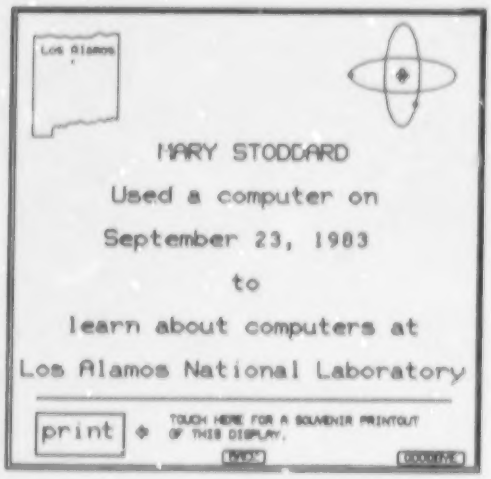


Figure 3.
Souvenir printout frames.

substantial savings in cost and time which result from using superfast computers to do the large computational problems facing scientists.

BEST COPY AVAILABLE

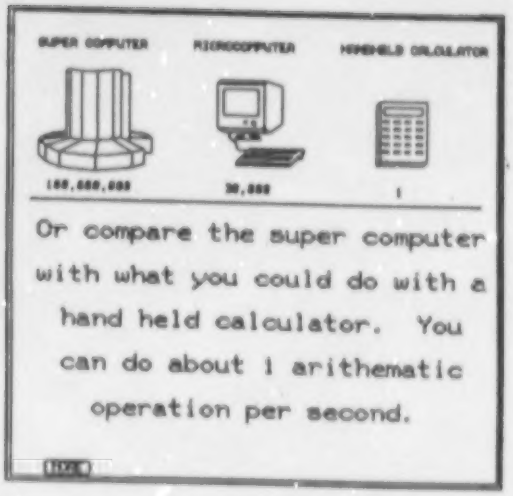

SUPER COMPUTER
MICROCOMPUTER
HANDHELD CALCULATOR
100,000,000
30,000
1
Or compare the super computer with what you could do with a hand held calculator. You can do about 1 arithematic operation per second.

Figure 4.
Comparison of speed of calculations.

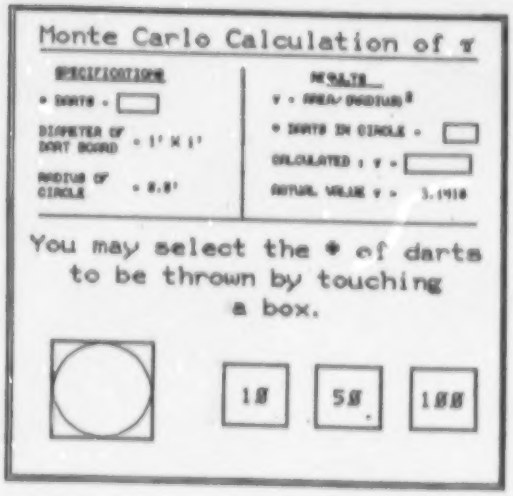

Monte Carlo Calculation of π
SPECIFICATIONS
RESULTS
DIAMETER OF DART BOARD = 1' X 1'
RADIUS OF CIRCLE = 0.5'
You may select the # of darts to be thrown by touching a box.
10
50
100

Figure 5.
How simple computations can require much arithmetic.

Visitor Reaction

The exhibit has been in use at the Bradbury Science Museum since April, 1983 and has received heavy usage. Although surveys are the most common evaluation methodology in museums (Yellow et al, 1979), informal observations have revealed very useful information (Loomis, 1977). The evaluation methodology chosen here was observation. The observer randomly selected a visitor to follow and informally watched the visitor interact with the exhibit. The observer had a clipboard, containing a checklist for each visitor. The checklist contained questions regarding: demographics, time spent on exhibit, visitor interaction with others, nonverbal visitor reactions, and behavior regarding the souvenir printout. Results of the observations have indicated that if visitors become attracted to the exhibit, they usually stay and become very involved. Few complications in usage of the program were observed. Only one software change has been necessary. This change allows visitors to generate multiple souvenirs. It was required because often several visitors would proceed through the exhibit together, and each desired a souvenir. The touch capability has proven extremely popular, as is the Spanish translation option. Cognitive gains as a result of the exhibit are difficult to measure in informal learning situations such as this, yet discussions with a sampling of visitors following exhibit interaction suggested that the objectives of the project were met.

Because of the popularity of this exhibit, another CD 110 based exhibit is being developed to present research applications from elsewhere in the Laboratory.

Summary

Constraints of the museum environment present unique challenges to the instructional designer. The project required a brief program with a high degree of interaction, no keyboard, a design responsive to a diverse target population and a balance of instruction with fun -- all in a sophisticated subject matter area. The development investment was high, with over 100 hours of design time and 60 hours of programmer time.

The goals of this exhibit seemed unrealistic in light of the constraints, yet visitor reaction has indicated a high degree of success. The exhibit has shown that microcomputer-based exhibits are effective and adaptable. In spite of the high development costs, the large number of visitors to the museum each year result in this being a cost effective exhibit.

BEST COPY AVAILABLE

Acknowledgements

The following individuals were also instrumental in conceiving and implementing the microcomputer exhibit: Bob Hiromoto, Ira Agins, Ray Elliott, and Chris Caramana.

References

(1) Kimche, L. "Science Centers: A Potential for Learning," *Science*, Vol. 199, 20 January 1978.

(2) Laetsch, W. M., "Highlights of the International Conference of Science and Technology Museums," *Proceedings from the Association of Science - Technology Center Conference*, 1976.

(3) Loomis, R. J. "Getting to Know the Audience," *Longwood Program Seminars*, 8, 1977.

(4) Miles, R. S. "Human Biology and the New Exhibition Scheme in the British Museum." *Curator* 21.1, 1978.

(5) Patterson, R. E. "Computer Aided Instruction in the Museum Environment." *Proceedings of the 24th International ADCIS Conference*, May 1983.

(6) Yalow, E. S., Strossen, L. L., Jennings, E. L., and Linn , M. C. "Improving Museums Through Evaluation." Unpublished manuscript. Stanford University, 1979.

BEST COPY AVAILABLE

The Coming of Age of Educational Technology: Meta-Analyses of Outcome Studies of Computer-Based Instruction

Robert L. Bangert-Drowns, James A. Kulik, & Chen-Lin C. Kulik

The University of Michigan

ABSTRACT

The first major applications of scientific technology to education were made by psychologist B. F. Skinner three decades ago. In the years since, the emphasis in instructional technology has shifted from programmed instruction to individualized systems of teaching to computer-based instruction. Using a technique called "meta-analysis," researchers have pooled results from hundreds of individual studies to draw conclusions about the effectiveness of these instructional technologies.

The meta-analytic reviews show that computer-based approaches to teaching have been especially effective. The record of CBI compares favorably to those of programmed and individualized instruction. Four other conclusions can be drawn from the meta-analytic reviews. First, CBI has been most effective in improving student achievement at the precollege level. Second, CBI has had positive effects on student attitudes toward subject matter, toward instructional quality, and especially toward computer technology. Third, CBI has yielded substantial savings in instructional time. And finally, the positive findings on CBI outcomes are very robust, emerging consistently in study after study in spite of methodological differences and differences in study setting.

* * * * * *

Thirty years ago B. F. Skinner wrote a striking prescription for education's ills. His article "The Science of Learning and the Art of Teaching" called for a technological revolution:

> We are on the threshold of an exciting and revolutionary period, in which the scientific study of man will be put to work in man's best interest. Education must play its part. It must accept the fact that a sweeping revision of educational practices is possible and inevitable.

In retrospect, Skinner's words seem prophetic. In the last three decades, psychological study and technolgical innovations have come to influence more and more the way we teach our children. But oddly enough, Skinner's prophecies do not seem dated today. In spite of the progress of the past 30 years, we still are on the threshold of the technological revolution.

What Skinner hoped for thirty years ago was not the complete automation of teaching. Skinner realized that teaching is an essentially human art, resting on human creativity, motivation, and interaction. But while teaching requires a human touch, it must also respect principles of learning that can be characterized scientifically. These four principles must constitute the core of any application of educational technology. First, the learner must be allowed to be active. Second, the learner's activity must be carefully guided by frequent and immediate reinforcement of his or her correct responses. Third, the material to be taught must be presented in graduated sequence, from the most simple and general to the most complex and detailed. Finally, each learner must be allowed to control the the rate at which he or she progresses through the material.

Skinner thought that a teaching method consistent with these principles could be easily devised. He championed the use of programmed instruction. His teaching

machines typically presented single sentences of the fill-in-the-blank variety. Students progressed at their own rate from sentence to sentence, correcting their own answers immediately as they went. Understanding the sentence in each frame depended on mastery of previous frames and students moved to increasingly complex materials.

Educators soon began noticing flaws with this kind of presentation. Lessons were broken into what seemed to be trivially small parts presented in an inflexible and monotonous sequence. Learners became bored with learning. Disenchantment grew, and within a few years the teachers who had originally embraced programmed instruction were turning away from it.

From the disenchantment with programmed instruction, a new technology emerged: individualized instruction. It came in a variety of forms: Individually Prescribed Instruction, Project PLAN, Individually Guided Education, Keller's Personalized System of Instruction. These teaching systems share many features. Like programmed instruction, individualized instruction emphasizes self-pacing and the achievement of mastery. But individualization improved on programming by using larger units of instruction, sometimes called learning activity packages or modules; by giving learners freedom to choose different means of mastering specified objectives; and by actively encouraging greater peer and teacher interaction. Although this teaching approach has influenced the design of instructional materials in recent years, individualized instruction has not affected teaching as dramatically as its proponents hoped it would.

With computers nearly omnipresent in society today, schools are naturally eager to prepare children for the computer age. To some, use of the computer in ordinary school work would constitute the best possible preparation for use of the computer in adult work. More, the computer fits Skinner's prescription for scientific instruction better than the earlier technologies do. The computer can require the student to respond actively, reinforce correct responses immediately, work at the student's rate and adjust the level of instructional complexity, follow a systematic instructional plan, and provide both primary instruction and remedial review. The computer can be an infinitely patient tutor, a scrupulous examiner, an engaging performer, and a tireless administrator in the classroom. With the advent of the microcomputer, all this can be bought at a much reduced price and with greater flexibility.

Do we have evidence to justify the use of computers in education? Why should the taxpayer believe that any new expense will bring better results than the individualized, programmed, or even traditional instruction that came before it? Why should teachers be willing to welcome yet another instructional innovation? To answer these questions, we could put the computer to the test. We could divide our students into two groups, give one group instruction by computer and the other instruction by traditional methods. Then we could let both groups take the same final exam and see who does best.

This has in fact been done innumerable times. None of the studies, however, has yielded a definitive conclusion. Findings vary from study to study, and seem to vary in complex ways as a function of study setting, software, hardware, and students. The question therefore remains: Is there any way to characterize the effectiveness of computer-based instruction in general?

At The University of Michigan, my colleagues and I have used a technique called "meta-analysis." We began by collecting every study we could find that compared computer-based instruction to traditional instruction. We carefully screened the studies, keeping only those that had no major methodological flaws and that reported outcome data in quantitative terms. Using certain algebraic manipulations, we reduced the outcomes of each study to a common scale, called an effect size. The effect size for each study describes the different outcomes of the experimental and control groups in standard deviation units. Next we coded the features of each study in categorical or quasi-quantitative terms. Finally, we looked for relationships between study features and effect sizes.

Our team at The University of Michigan recently completed two meta-analytic reviews of findings on computer-based instruction--one focusing on secondary school teaching and one on college-level teaching--and we are currently finishing a third on elementary school instruction. We are not the only ones, however, to conduct meta-analytic reviews of computer-based teaching. Another meta-analytic review was produced by a researcher at the University of Colorado in 1978 and still another was done at the University of Iowa in 1981. The findings I will report therefore have been verified by a number of researchers, pooling results from about 150 separate studies.

Based on the available research, our first observation is that computer-based instruction has been used most effectively in the elementary and secondary schools, less so in colleges. This is perhaps because software developed before the time of this research exploited the computer capacities most adapted to lower level learning: its patience in drilling and tutoring students and its capacity to respond immediately to student answers. The typical application of CBI in precollege grades raised an average student's achievement scores by about .4 standard deviations, or from the 50th to the 66th percentile. This is a substantial gain. The achievement test score that a CBI college student might expect is .25 standard deviations higher than a traditional student's score. This difference is equivalent to the difference between the 60th and 50th percentiles, a noteworthy although smaller gain.

Our second observation is that CBI improves student ratings on attitude measures. Students taught by computers generally rate the subject matter that they are studying about .15 standard deviations higher than other students rate the same material taught traditionally. This is true at all grade levels. When college students were asked to rate the method of instruction in their class, the CBI students rated their instruction about .24 standard deviations higher than the other students rated traditional instruction. Most interesting of all, we found four studies at the secondary school level that examined student attitudes toward computers. Students who used CBI had attitudes that were .61 standard deviations more positive than the attitudes of children who did not use CBI. This is a large difference. Computer usage in instruction seems to prepare students functionally and attitudinally for the computer culture in which they will live and work.

Our third major conclusion is that computer-based instruction can produce substantial savings in instructional time. In two studies at the secondary school level, savings in instructional time were 39% and 88%. In eight studies at the college level, the average conventional course required 3.5 hours of instruction weekly; the computer-based approach required only 2.25 hours. There is little doubt that students can be taught with computers in less time than with conventional methods.

Our fourth major conclusion is that findings on CBI outcomes are very robust. Differences in the instructional environment, the samples of students used, and the methodological design of the study had no substantial effects on the outcome of the instruction. We may have to place two limits on the generality of conclusions about CBI, but even these restrictions are tentative because the relevant factors have been examined in very few studies.

One thing that may limit the effectiveness of CBI is its use as a complete replacement for conventional instruction. CBI is not often studied in this way but the few studies where CBI is the only means of instruction show unimpressive findings. Total reliance on the computer as teacher seems to be one thing schools should avoid. The effectiveness of CBI appears to be enhanced, on the other hand, by the use of up-to-date programs and computers. Studies carried out in the period from 1976 to 1980, when software and hardware had increased in sophistication over earlier generations, showed achievement gains almost twice as large as earlier studies. This gives us reason to hope that even greater gains can be achieved with the programming inventiveness of the future.

CBI has proven itself against conventional instruction. In addition, CBI compares favorably with the two earlier technologies, programmed instruction and individualized instruction, especially at the precollege level. At this level, programmed and individualized instruction produce effect sizes of only .1 standard deviations, representing a change in achievement test scores from the 50th to the 54th percentile. Computer-based teaching showed an average effect size of .4 standard deviations, increases from the 50th to the 66th percentile. At the college level, CBI is just as effective as programmed instruction, though not quite as effective as individualized instruction.

The available research attests to the potency of computer use in the classroom. It is important to remember that these findings have been gathered from the past. If computers have already produced such a commendable record in instruction, one can only be excited by the future. Given the increasing usage in society at large, an expanded role for the computer in instruction appears inevitable. We should reassess our commitment to the basic principles presented by Skinner and elaborated in the thirty years since. Particularly, we first need to show our students that while learning is ultimately their responsibility, we still hope to provide an educational environment suited to their unique capacities, limitations, and motivations. Second, we need to reassure teachers that we respect their craft as a vital human enterprise. Given these conditions, we can only be

optimistic about educational technology's coming of age.

* * * * * *

REFERENCES

Burns, P. K., & Bozeman, W. C. (1981). Computer-assisted instruction and mathematics achievement: Is there a relationship? Educational Technology, 21, 32-39.

Hartley, S. S. (1978). Meta-analysis of the effects of individually paced instruction in mathematics. Dissertation Abstracts International, 38(7-1), 4003. (University Microfilms No. 77-29, 926)

Kulik, J. A., Kulik, C.-L. C., & Cohen, P. A. (1980). Effectiveness of computer-based college teaching: A meta-analysis of findings. Review of Educational Research, 50, 525-544.

Kulik, J. A., Bangert, R. L., & Williams, G. W. (1983). Effects of computer-based teaching on secondary school students. Journal of Educational Psychology, 75, 19-26.

Kulik, J. A., & Bangert-Drowns, R. L. (1983). Effectiveness of technology in precollege mathematics and science teaching. Journal of Educational Technology Systems, 12(2), 137-158.

Skinner, B. F. (1954). The science of learning and the art of teaching. Harvard Educational Review, 24, 86-97.

Using Interactive Logo Procedures

to Enhance Clinical Interviews with Children

Larry Flick

Park Tudor School

Abstract

This is a report of a pilot effort to develop a method for interviewing children with the aid of an interactive computer program. Reported here are the results of several group sessions with children of elementary school age and one in depth interview with a 14 year old girl.

The computer language Logo was used to simulate the application of forces on an object moving on a frictionless surface. Following work that has already shown the ability of specially designed Logo procedures to reveal intuitive thought (deSessa, 1982; White, 1983), this study combined the clinical interview with Logo driven problems. Utilizing the list processing features of Logo, the child's interaction with the computer was captured and analyzed in conjunction with a tape recorded interview.

The child's keyboard responses to force and motion problems provided an analytic framework from which her concepts were interpreted. It also showed her pattern of dealing with new information. The basic problem was to make the turtle move under a simulation of Newton's First Law across a circular target. The keyboard responses were analyzed according to the velocity of the turtle, turtle heading, bearing and distance of the target from the turtle, and the time between and sequence of commands.

Statements by the child were cross validated with the record of keyboard responses. The computer record was also used to raise questions concerning conceptual points of view not stated by the child. Implications for instruction are discussed with an emphasis on how Logo procedures can be used to enhance the child's ability to understand physical laws and the teacher's ability to understand the child's mode of thought.

Purpose

To glean from individual children their intuitive perspective from which they approach a force and motion problem.

To investigate the patterns of behavior that result from the addition of new information to a problem solving situation.

To develop the use of interactive computer programs as a method for recording and analyzing problem solving behavior.

To evaluate the strength and weaknesses of computer simulations as a tool for giving expression to children's concepts.

Interest in describing the cognitive structures of children has been stimulated by studies that show children hold science concepts that differ from the accepted "adult" science even in the presence of relevant instruction (Nussbaum & Novak, 1976; Nussbaum & Novick, 1980; Shavelson, 1972; Siegler, 1978; White, 1983). One approach at description has focused on the competence of children to reason logically. These studies have typically employed written tests whose items vary on dimensions of complexity, familiarity, symbolism, and type of logical principle (O'Brian &

Shapiro, 1968; Roberge, 1970). This perspective has been confounded by the result that many problems children have with logical reasoning are in common with adolescents and adults (Ennis, 1975; Ennis, 1978; Lawson, Karplus, & Adi, 1978).

By attempting to follow the many possible lines of deduction a child can make in a problem, our purpose is more in keeping with the early work of Jean Piaget (1959; 1967). In this line of research the objective is to follow dynamic lines of reasoning in a clinical interview format. We will follow the strict guidelines as described by Piaget (1967), but our purpose will be different. Rather than attempting to assess global rules of thought, we will seek a relationship between what the child believes is the problem, what she thinks is necessary for its solution, and the way in which she uses information toward the solution.

The interview as a tool for revealing a child's thought process is problematic due to the non-verbal nature of her thinking. Piaget (1959) describes the child's thought as ego-centric. It is characterized by the dominance of visual schema that leads to the resolution of opposing or contradictory data without analysis. Fischbein (1975) describes this thought as intuitive, i.e. patterns of thought which are closely tied to action and broadly stabilized by repeated contact with reality. In our study we view the computer record as an observation that reflects intuitive, non-verbal thought. Interaction with the keyboard is a physical action that is not necessarily mediated with a verbal mode of analysis. It is this that enhances the clinical interview while maintaining the theoretical perspective from which it came.

The computer simulation of the physics problem was inspired by Seymour Papert's (1980) vision of physical objects as cognitive models. Papert describes how watching gears in his childhood became a tool for understanding mathematics in later years. He called them math-like objects. They gave expression to mathematical ideas that were unexpressible in childhood. The Logo (Davidson, 1982) turtle under the control of Newton-like laws is intended to be such an object for our subjects. This study is an attempt to use the computer to develop a physical mechanism to aid children in expressing ideas that are otherwise acccessible only at an intuitive level. This project is an extension and modification of the work of Andrea diSessa (1982).

Methods

The clinical interview as conceived by Piaget (1967) formed the framework for our collecting of data. The adoption of this qualitative approach stems from our goal of observing the idiosyncratic behavior of individual children. Objective, quantitative methods are not dynamic and therefore cannot follow the many points of view offered by a single subject (Wilson, 1977). At the same time qualitative studies offer many threats to validity (McCall, 1969). We attempt to meet those threats by observing specific guidelines in conducting the interview. The computer record complimented the interview data by both validating conceptual frameworks interpreted from the interview and offering points of view not expressly stated in the interview.

The Interview

The interview will be recorded and conducted before, during, and after the interaction with the computer simulated problem. The fundamental guidelines for the interview are as outlined by Piaget (1967):

1. The form and substance of questions are to be modeled after questions spontaneously asked by children.

2. The interviewer must be at all times be sensitive to answers that have possibly been dictated or suggested by the question or its contenxt.

3. The interview questions will seek to uncover uniformity accross children's or within children's responses.

A general introduction concerning forces and motion preceded involvement with the computer. The child was asked to consider the effect of kicking an object on a very smooth, slick surface. If there were no walls how might the object be stopped? Can the object be stopped, assuming you could stay close enough to keep kicking it in various directions? There were no attempts to resolve problems or misconceptions. The computer simulation involved the Logo turtle being "kicked" by applying a "force" in any one of 12 possible directions. The turtle is an ideal Newtonian object in that once in motion it stays in motion until it reaches the out of bounds area of the screen. Crossing the edge of the screen caused the program to pause and restart upon pressing the return key. The goal for causing the turtle move was to make it cross a circular target.

The Computer Simulation

Each time the "P" key was pressed the turtle rotated in a clockwise direction by 30 degrees; similarly for left rotations with "Q". A kick was applied by pressing the spacebar. A space produces a change in the incremental value added to the X position of the turtle by the sine of the turtle heading times a force variable. The incremental Y value was changed by the cosine of the heading times force. In that the turtle can only turn in increments of 30 degrees, the position of the target at a 45 degree heading from the turtle's starting point makes an initial aim and shoot strategy unsuccessful.

Each single key stroke command is recorded via the list processing features of Logo. This list is recorded on disk at the completion of each run at the target. A "time" counter in the command procedure keeps track of the amount of motion between commands so that a precise replication of turtle behavior can be made.(see Fig. 2)

Intuitive thought can solve this computer problem. Groups of children operating freely with a form of the program in this study show that by about age 11 children can master the problem in one 30 minute session. As a result of this experience the program used in this initial interview increased the force variable by 1 with a press of the game paddle 1 button by the experimenter. A faster moving turtle forces a reevaluation of strategies.

```
SETUP  :FORCE :TRIAL# :RECORD
   INITIALIZE
   FORCE.PROBLEM
   WRITE.TRIAL.TO.DISK
   INCREASE.FORCE?
   SETUP  :FORCE :TRIAL# + 1 []
END

FORCE.PROBLEM
   COMMAND READKEY
   MOVETURTLE
   FORCE.PROBLEM
END

COMMAND  :KEY
   INCREMENT.TIME
   IF NO.COMMAND [STOP]
   TRACK.COMMAND.AND.TIME
   EXECUTE.COMMAND
END
```

Figure 2: Toplevel structure of the interactive program.

Data Analysis

The raw data comes in two forms. There is the interview transcript and the Logo list of commands with the time intervals between them. The transcript was written in roughly propositional form in such a way that the subject's expressions could be coordinated with that part of the computer activity that originally accompanied it. The computer record was recorded by trials corresponding to each time the turtle ran out of bounds. A trial, Tn, is in the form of a list such as:

[1 69 P 8 K 14 P 8 K 42 P 4 K 3 Q 12]

This list of clockwise and counter-clockwise rotations and kicks has as its first element the force variable value. Each of these trials, Tn, becomes the input to a program that analyzes the turtle state just prior to a command. The output (see Fig. 3) of this turtle state procedure is a statelist of lists:

[[state prior to command - P][state prior to command - K]. . .[state prior to command - Q][state immediately after Q is executed]].

Each element of the statelist is itself a 7 element list:

[SP DIR.T HEAD.V HEAD.T DIST TIME COM]

The elements of the statelist are defined as follows (see Fig. 1):

SP = speed of turtle in # of screen locations skipped each time the command loop is entered.

DIR.T = direction of turtle velocity with respect to target.

HEAD.V = heading of turtle with respect to the velocity.

HEAD.T = heading of turtle with respect to the heading towards the target.

DIST = distance from the current turtle position to the target.

TIME = loop counter of turtle movements since the last command.

COM = keyboard input: P, Q, or a space places a K on the list.

The relationship between and among these variables can be used to represent patterns of keyboard responses. For example, if a strategy of repeatedly aiming for the target and kicking is used the value of HEAD.T will be low before each kick. The notion that a kick must be applied as soon as the turtle is headed in the desired direction would be

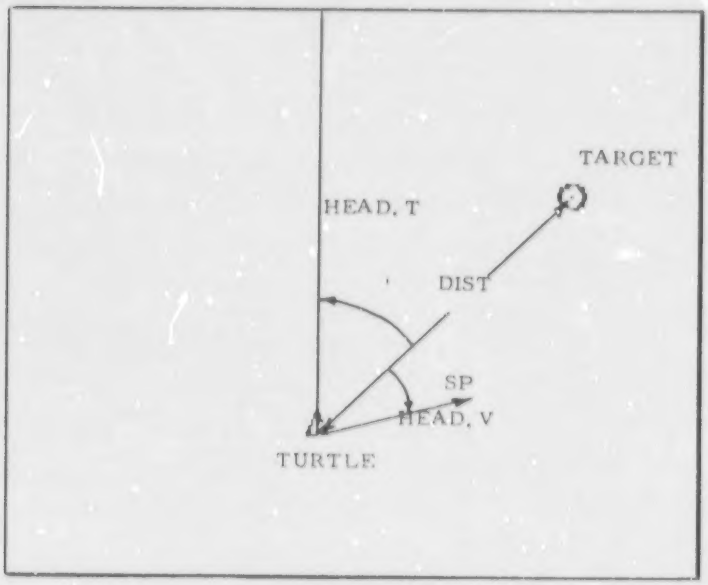


Figure 1: Variables associated with the analysis of the turtle state after each command.

represented by a short TIME interval between a kick and the preceding turn. Toward this end another list processing routine reads each trial list, Tn, and records the intervals between 2 commands, e.g. between P and K. The output is an interval list of which the following is an example:

[PK 12 PK 2 KP 10 PK 5]

Intervals from P to K and from K to P are recorded since we may also be interested in intervals between K and K or P and P. The subjects interest in speed might be represented by K-K intervals. In general the time between commands could be related to the amount of thinking going on in anticipation of a move to make the turtle head toward the target. The statelist analysis adds an asterisk before each command that is given while the turtle is on a trajectory that would cross the target.

LEE'S TRIAL T14
FORCE = 1

SP	DIR.T	H.V	H.T	DIST	TIME	COM
0	-45	0	-45	141	69	P
0	-45	30	-15	141	8	K
1.	-17	0	-17	128	14	P
1.	-18	30	12	120	8	K
1.9	- 8	15	7	39	42	*P
1.9	-10	45	35	32	4	*K
2.7	6	30	36	24	3	*K

Figure 3: Sample turtle state analysis

While most of these uses of the analysis procedure have not been fully explored with children, these comments are added here as a stimulus for further work. Exploring the richness that interacting with the computer can offer during a clinical interview will be the purpose of our next research efforts.

BEST COPY AVAILABLE

BEST COPY AVAILABLE

Results

This technique has been used with small groups of children grades 1 through 6. They all had had an introduction to the turtle graphics features of Logo and in general were familiar with computers and keyboards. Each child sat at his/her own computer and were instructed to guide the turtle across the target.

The group sessions, which were no more than 20 minutes of spontaneous interaction, showed a marked difference by age in the ability to control the turtle. Few if any in the 1st and 2nd grades actually grasped the nature of controlling the turtle. Many in these groups began to play with the turtle in other ways or showed disinteres in the idea altogether. In retrospect this behavior is perhaps not too different from what would happen if this same group were instructed to kick a soccer ball to a goal after being placed in the middle of a large playing filed. Controlling a ball over long distances would be for some quite tedious at best.

Some children simply enjoyed turning the turtle around repeatedly. Another said, "Press the spacebar, it will get faster, then it won't get boring." But in a deeper consideration of motion one asked, "Why does it keep going sideways?" Their efforts to get the turtle to go in the direction of its heading were failing. Another, after expressing similar frustration with the resultant vector said, "It just goes the way it was going (after giving it a turn and a kick)." Later he said, "You have to press it (spacebar) lots of times, then it will go the way it is pointing."

Although in these sessions no computer record was being taken, the keyboard behavior of children as young as 6 or 7 may offer some insights into the nature of their play. There were 2 efforts at real control. One 1st grader said she kept it going slow because she couldn't control it going fast. The other held his fingers in a horizontal position at the level of the target in order to execute a kick when the turtle was at that level. His strategy required that the turtle heading point to the right side of the screen early in its motion so that attention could be paid to the time to initiate the kick.

There was a tendency to overpower the unwanted turtle velocity with many kicks in the desired direction. This spontaneous strategy of more kicks implies faster implies proper direction can be thought of as an intuition born of experience. If the kicks were always interspersed with turns, one might wonder at the incorsistency with peddling a bicycle, pushing a wagon, or rolling an object down a hill. At the very least this strategy of overpowering the turtle suggests a dissociation of force from guidance in daily life.

The common expectation of the turtle heading determining the direction of velocity is also an intuition grounded in reality. Small objects under the control of children, such as balls, toy cars, and riding toys yield readily under their forces. The fact that other intuitions associated with larger objects or objects moving over larger distances, do not seem to come into play suggests that these have not had an environment which has stimulated much thought. It is also very likely that this result is an artifact of the smallness of the turtle and its association with smaller objects. Even so, this suggests that small objects in the classroom used to teach the nature of force do not elicit the desired thoughts nor aid in the production of new intuitive forms.

A 6th grader took the suggestion of kicking against the velocity in order to try to stop the turtle. The success that resulted immediately spawned a rule. The rule was based on this one specific event: Kick it 3 times, hit Q 6 times, and hit the spacebar 3 times. This discovery and the rule was passed around the room quickly. The rule was never generalized verbally but at least three 6th graders were able to stop the turtle in the target circle or anywhere else they chose.

The background of the formation of this stopping strategy suggests that this knowledge of forces that start and stop motion do not quickly grow beyond the specific situation that produced the idea. It is our contention from years of teaching this age child that a verbally oriented test on forces stopping motion would show a strict adherence to the rule stated at the time of the discovery. We also sensed that those children who received the rule verbally, even after trying it out were at a disadvantage until they tested the parameters of the rule.

The computational environment under discussion here offers a way of investigating the child's intuitive efforts at testing, whether expressed or unexpress, after the addition of new information. As educators we would like to enhance the learning environments to promote more fluent and rigorous testing of ideas.

A Clinical Interview

Lee: I gave it a different command right then and. . .so I pressed the key and then. . .I don't know, but when it stopped you (I) can give it a different command.

Experimenter: You can give it a command anytime you want to.

Lee: Yes, but I mean you can change its direction.

The initial interpretation of the kick command set the conceptual stage for this pilot interview. Lee's first 5 tries at the target produced no change in direction after the initial kick. At T6 she tried a second kick during the course of her turning. She kicked it while the turtle was at a heading of 150 degrees with respect to the target, as if the idea of trying this out replaced the goal of trying to cross the target. This was a significant event for it initiated the strategy of using the kick to change direction.

Her initial concept was related to driving a car (not something this 14 year old girl had ever done). The records of her trials show repeated use of P-Q sequences. Turning right then left would seem to have no value in this problem and, of course, it led to no success with the target. Her first 6 trials had 14 such moves in them. She explained:

it back again with her arms and hands) you can hit it (make the turtle change directions a small amount).

From T7 on there were only 2 P-Q sequences. There seemed to be an effective conceptual change due to an environment in which she could convince herself of the worth of the new point of view.

T12 through T15 produced a series of 4 crossings of the target. She offered little in the way of explanation as to how she did this. Her computer record shows that she repeated the same strategy each time with a decreasing in number of commands.

Trial#	#commands	Total time
12	7	79
13	8	70
14	4	30
15	4	35

While replaying these successes she was convinced that the correct strategy was to point the turtle at the target and kick it. The kicks during these 4 trials were always applied with HEAD.T < 40 degrees, with most of them < 15 degrees. Within the 30 trials during this interview, Lee made 83 kicks with only 13 of them applied at HEAD.T > 40 degrees. With the playback presentation of the turtle motion contradicting to this strategy, showing that "winning" kicks were applied at headings not directly pointing toward the target, she held her ground. In retrospect we wonder if she could have utilized a way of bringing in intuitions concerning curvilinear motion.

Significance

The ability of educators to regularly investigate children's cognitive structures in relation to subject matter structure will be affected by the success of the type of tools developed in this study. Results which show that college freshman know a lot of physics but are unable to relate it to their world (Gunstone & White, 1981) highlight the disparity between school learning and day to day thinking. Perhaps, as Seymour Papert (1980) suggests, our world is poor in systems but rich in specific knowledge. Rodrigues (1930) observed that while by 12 years of age the quality of scientific explanations approached that of adults, much younger children use statements of the nature of physical law consistently. We observed 6th graders formulating rules concerning force spontaneously. What is needed are environments that nurture the use and generalization of this kind of thinking.

The use of the computer as a thinking tool will depend upon the development of interactive programs that can sense frustration and play. Through a deeper knowledge of the alternative points of view from which a problem can be viewed, we can build more enticing environments for exploration. An advantage to a computer environment over a more natural one is that the relevant factors are not only known by the educator but can be limited in number and effect. The systematic manipulation of factors of a problem in a dynamic computational model holds the promise of bringing intuitive ideas to the level of expression.

References

Davidson, L. J. Apple Logo reference manual. Lachine, Quebec Canada: Logo Computer Systems, Inc., 1982.

deSiasa, A. Unlearning Aristotelian physics: A study of knowledge-based learning. Cognitive Science, 1982. 6, 37-75.

Ennis, R. H. Children's ability to handle Piaget's propositional logic: A conceptual critique. Review of Educational Research, 1975, 45, 1-41.

Ennis, R. H. Conceptualization of children's logical competence: Piaget's propositonal logic and an alternative proposal. In L. S. Siegel & C. J. Brainerd (Eds.),Alternatives to Piaget: Critical essays on the theory. New York: Academic Press, 1978.

Gunstone, R. F., & White, R. T. Understanding of gravity. Science Education, 1981, 65, 465-478.

Lawson, A. E., Karplus, R., & Adi, H. The acquisition of propositional logic and formal operational schema during the secondary school years. Journal of Research in Science Teaching, 1978, 15, 465-478.

McCall, G. Data quality control in participant observation. In G. McCall & J. L. Simmons (Eds.), Issues in participant observation. Reading, Mass.: Addison Wesley, 1969.

Nussbaum, J., & Novak, J. D. An assessment of children's concepts of the earth utilizing structured interviews. Science Education, 1976, 60, 535,550.

Nussbaum, J., & Novick, S. Brainstorming in the classroom to invent a model: A case study. Unpublished paper, Hebrew University, Jerusalem, Isreal, 1980.

O'Brien, T. C., & Shapiro, B. J. The development of logical thinking in children. American Educational Research Journal, 1968, 5, 531-541.

Papert, S. Mind-storms: Children, computers, and powerful ideas. New York: Basic Books, Inc., 1980.

Piaget, J. The language and thought of the child. (Marjory & Ruth Gabain, trans.). London: Routledge & Kegan Paul Ltd., 1959.

Piaget, J. The child's conception of the world. (Joan & Andrew Tomlinson, trans.). London: Routledge & Kegan Paul Ltd., 1967.

Roberge, J. J. A study of children's abilities to reason with basic principles of deductive reasoning. American Educational Research Journal, 1970, 7, 583-595.

Rodrigues, D. M. Notions of physical laws in childhood. Science Education, 1980, 64, 59-84.

Shavelson, R. J. Some aspects of the correspondence between content structure and cognitive structure in physics instruction. Journal of Educational Psychology, 1972, 63, 225-234.

Siegler, R. S. The origins of scienctific reasoning. In R. S. Siegler (Ed.). Children's thinking: What develops. Hillsdale, N. J.: Lawrencel Erlbaum Assoc., 1978.

White, B. Y. Sources of difficulty in understanding Newtonian dynamics. Cognitive Science, 1983, 7, 41-65.

Wilson, S. The use of ethnogrpahic techniques in educational research. Review of Educational Research, 1977, 47, 245-265.

Skill Theory: The Process for Finding the Lost Content

Catharine C. Knight
University of Denver
Walter J. Kuleck
Cognitive Processes Inc.

ABSTRACT

Much of educational programming, both computer based and traditional, is focused upon "getting the right answer" rather than on comprehending and enhancing the learning process itself. Until recently, the development of process oriented educational programming has been inhibited by the lack of an appropriate, well documented, practical framework upon which to base such programming. Skill theory (Fischer, 1980) is suggested as one theoretical ground plan that is particularly adaptable to computer based instruction. A skill theory based schema of beginning reading skills is presented as an illustration of the practical application and usefulness of Fischer's formulation. Children's responses to tasks derived from the skill theory schema were processed by means of order analysis (Krus, 1976). The findings of order analyses may be graphically represented as a form of flow charts similar to those used in designing computer programs. Hence the results of skill theory analysis may be expressed in terms of computer paradigms. The complementary interaction of skill theory and computer based instruction is discussed in terms of the planful, efficient presentation of material, its potential for effective teaching and remediation, and the opportunity it offers to learn more about learning in the process.

INTRODUCTION

When a learning child responds with an appropriate answer, this product of his or her learning is clearly evident and easily evaluated. The process of learning is not so observable. That is, the steps the learner goes through in order to respond correctly -- the tasks mastered along the way, the cognitive skills utilized, the false starts and unfruitful digressions -- are not so easily recognized, much less assessed (Watt, 1983). Consequently, it is not surprising that most learning aids, programs, and related materials are focused on "getting the right answer," rather than on comprehending and enhancing the learning process.

Until recently, the development of process oriented educational programming has been inhibited by the lack of an appropriate, well documented, practical and predictive framework upon which to base such programming. Educators in computer based instruction have been handicapped by the same difficulty hampering educators in general: the lack of a flexible and viable ground plan with rules of passage (Gibson & Levin, 1975).

While it is understandable that most educational programs, computerized or traditional, are product oriented, we feel that a conceptualization and modeling of the learning process is vital to the optimal development and utilization of computer based instruction in the classroom and in the home (Papert, 1980). Understanding of the learning process is mandatory in order to efficiently program even a simple learning pathway. The more a programmed progression mirrors the learning process, the more quickly and completely the learner will understand and assimilate the material presented (Gibson & Levin, 1975). Also, the learner will gradually become aware of his or her own efficient learning processes and potentially utilize this self knowledge (Flavell, 1977) to facilitate more efficient learning in the future.

Finally, process oriented learning, appropriately programmed and executed, is more fun. The learner simply has less struggle and more success.

SKILL THEORY

Fischer's (1980) conceptualization, called skill theory, describes how cognitive skills are actually built, diversified, and generalized. The theory attempts to explain and predict "developmental sequences in any domain [such as reading] at any point in development, by integrating behavioral and cognitive developmental concepts" (p. 477). Cognitive development was described "as the construction of hierarchially ordered collections of specific skills" (p. 477).

Fischer summarizes his theory in the following:

> Skill theory provides an abstract representation of the structures of skills that emerge in cognitive development, together with a set of transformational rules that relate these structures to each other. The structures and transformational rules comprise a tool for explaining and predicting the developmental sequences and synchronies from birth to adulthood...
>
> Here is a brief overview: Skills develop step by step through a series of... levels. ...The levels specify skills of gradually increasing complexity, with a skill at one level built directly on skills from the preceding level. Each level is characterized by a reasonably well defined structure that indicates the kinds of behaviors that a person (child or adult) can control at that level. The skills at each level are constructed by a person acting on the environment. S/he performs several actions induced by a specific environmental circumstance, and the way these actions occur in that circumstance provokes her/him to combine the actions: The person thus combines and differentiates skills from one level to form skills at the next higher level. The movement from one level to the next occurs in many micro-developmental steps specified by a series of transformation rules. Notice the skill develops through levels, not stages: Development is relatively continuous and gradual, and the person is never at the same level for all skills. The development of skills must be induced by the environment, and only the skills induced most consistently will typically be at the highest level that individual is capable of. Uneveness in development is therefore the rule, not exception. The level of skills that are strongly induced by the environment is limited, however, by the highest level of which the person is capable. As the individual develops, this highest level increases, and so s/he can be induced to extend these skills to the new, higher level. (pp. 479-480)

Fischer's skill theory has been subjected to test in several cognitive domains and has fared extremely well (Hand, 1981; Corrigan, 1983; Harter, 1983). It does indeed appear to be a predictable, appropriate, and highly generalizable representation of learning development.

Knight (1982) applied a skill theoretical approach to the development of beginning reading skills. Using semantic, phonological, and visual graphic domains (LaBerge & Samuels, 1974), she constructed a hierarchial schema or ground plan of beginning reading skills (Figure 1). One hundred and twenty primary level children were tested on seven tasks derived from the schema reflecting varying levels of beginning reading skill (Knight & Nelsen, 1982).

The results were subjected to order analysis (Krus, Bart & Airasian, 1975; Krus, 1976) as well as several other analytical techniques. Order analysis should be of particular interest to computer based instruction strategists, since its results may be expressed graphically in the form of a dendrogram, which is simply a form of flow chart. We submit that such flow charts may then be readily transformed into computer programming flow charts. Hence, the results of skill theory analyses, presented using order analysis models, may be readily translated into computer programming paradigms.

RESULTS

As portrayed in Figure 2, Knight's hypothesized order of skill development was strongly supported by the results of order analysis of the children's responses. In fact, all the children's response patterns were accounted for by Knight's schema (Figure 1). Moreover, the children's erroneous response patterns could also be traced through the ground plan and troublesome skill areas thus clearly identified.

Figure 1. A Schema of Word Reading Skills

BEST COPY AVAILABLE

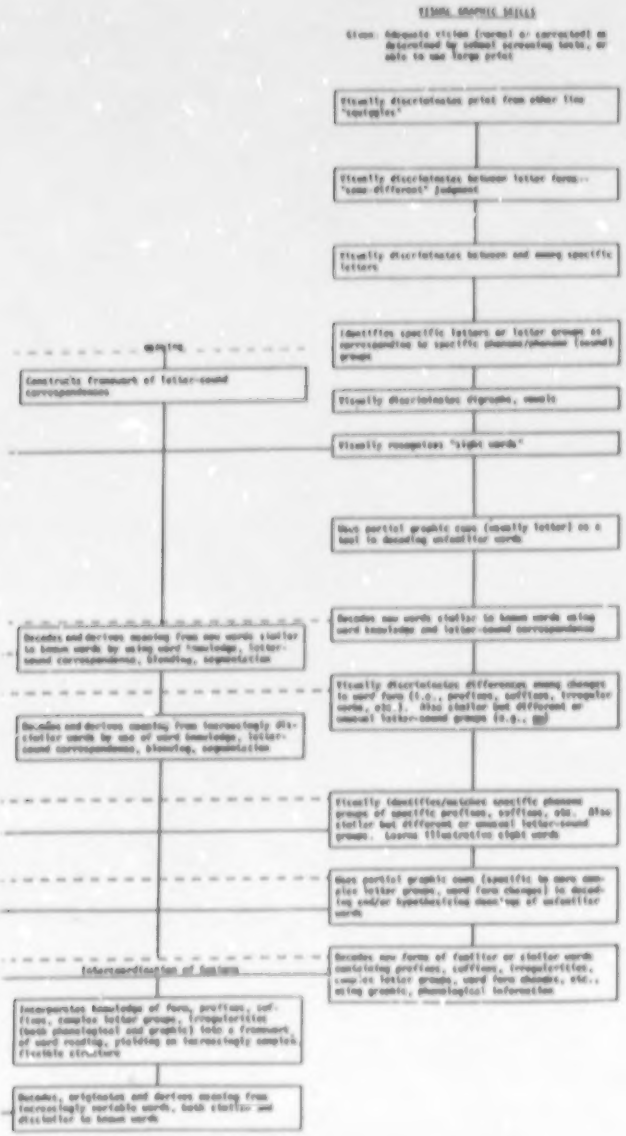

VISUAL GRAPHIC SKILLS
Given: Adequate vision (normal or corrected) as determined by school screening tests, or able to use large print
Visually discriminates print from other line "squiggles"
Visually discriminates between letter forms--"same-different" judgment
Visually discriminates between and among specific letters
Identifies specific letters or letter groups as corresponding to specific phonemes/phonemes (sound) groups
Visually discriminates digraphs, vowels
Visually recognizes "sight words"
Uses partial graphic cues (usually letter) as a tool in decoding unfamiliar words
Decodes new words similar to known words using word knowledge and letter-sound correspondences
Visually discriminates differences among changes in word form (i.e., prefixes, suffixes, irregular verbs, etc.). Also similar but different or unusual letter-sound groups (e.g., ea)
Visually identifies/matches specific phonemic groups of specific prefixes, suffixes, etc. Also similar but different or unusual letter-sound groups. Learns illustration sight words
Uses partial graphic cues (specific to more complex letter groups, word form changes) in decoding and/or hypothesizing meaning of unfamiliar words
Decodes new forms of familiar or similar words containing prefixes, suffixes, irregularities, complex letter groups, word form changes, etc., using graphic, phonological information
Interrelationship of Systems
Incorporates knowledge of form, prefixes, suffixes, complex letter groups, irregularities (both phonological and graphic) into a framework of word reading, yielding an increasingly complex, flexible structure
Decodes, originates and derives meaning from increasingly variable words, both similar and dissimilar to known words

Figure 1. (Cont.) A Schema of Word Reading Skills

BEST COPY AVAILABLE

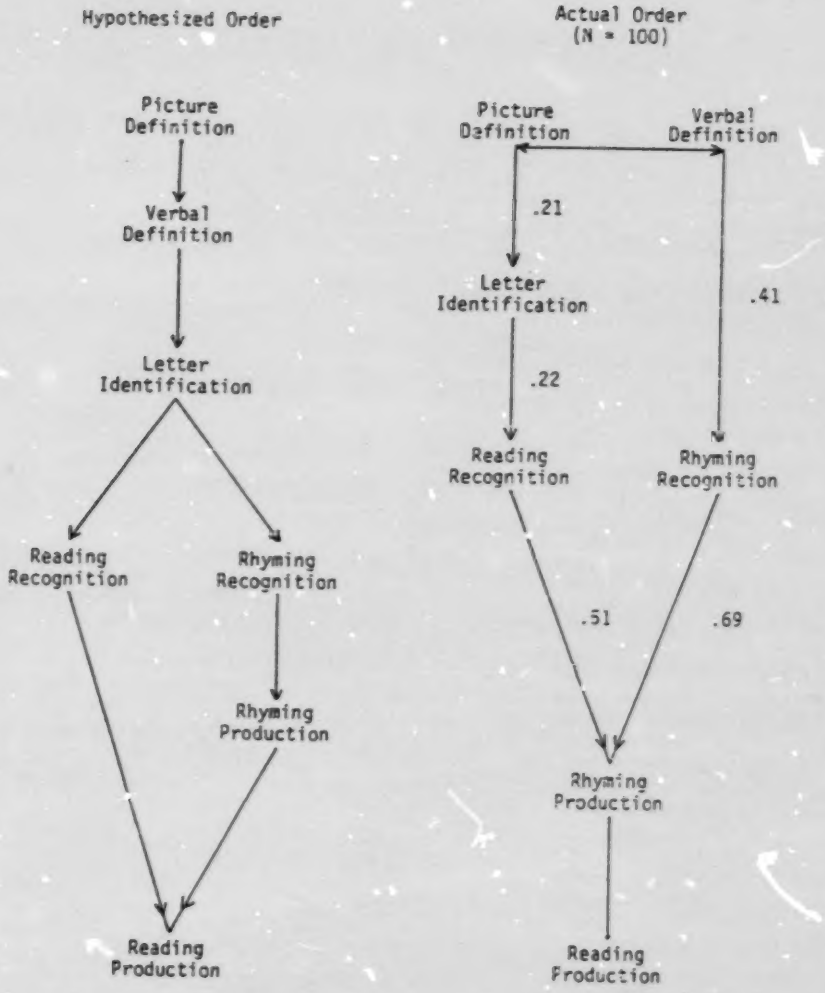


Figure 2. Hypothesized and Actual Orderings of Seven Subtasks Tested

BEST COPY AVAILABLE

A NEW WAY OF COMPUTER MODELING

We consider skill theory to be a new and promising way of computer modeling the learning process. Skill theory not only provides a way to sequence tasks or items, but also suggests how the skills might be presented. For example, letters must be related or "mapped" to their corresponding phonemes at a very basic recognition level (Knight, 1982; Knight & Nelsen, 1982), but at a more complex level, letter groups and sound groups must be intercoordinated and integrated to be successfully learned (Figure 1). The method of presentation of material at these two different levels would, according to skill theory, vary as a function of the transformation rule being employed for most efficient learning at each level. Hence, Fischer not only describes what is to be learned and at which level, but also suggests how material may be best presented for most efficient learning at each level.

Armed with such information, educators will be able to design instructional programs in a sound, planful, and efficient way. In time, it may not be necessary to utilize tradition and empirical trial and error as the primary strategies for designing and executing instructional material.

COMMENT

We expect that utilization of skill theory in conjunction with computer based instruction will have exciting and far reaching implications and applications. On one hand, skill theory provides a viable, testable way to model the process of learning, which can be applied to any content domain. Moreover, skill theory is a relatively complete representation of the learning process. Skill theory requires no conceptual or inferential leap from model to application. In other words, the gap between theory and empirical data is minimal.

On the other hand, computer based instruction appears to be an ideal medium with which to utilize skill theory as a way to improve the presentation and learning of instructional material. We expect that a complementary interaction of skill theory and computer based instruction will not only foster more efficient and planful instructional programming, but also provide a means to design and execute efficient and comfortable remediation.

Finally, the application of skill theory to computer based instruction will allow educators to learn more about the process of learning itself. Hence the benefits to educational researcher, educator, and learner are self perpetuating and self regenerating.

REFERENCES

Corrigan, R. (1983). The develoment of representational skills. In K. W. Fischer (Ed.), Levels and transitions in children's development. San Francisco: Jossey-Bass.

Fischer, K. W. (1980). A theory of cognitive development: The control and construction of hierarchies of skills. Psychology Review, 87, 477-531.

Flavell, J. (1977). Cognitive development. Englewood Cliffs: Prentice-Hall.

Gibson, E. J., & Levin, H. (1975). The psychology of reading. Cambridge: The MIT Press.

Hand, H. H. (1981). The development of concepts of social interaction: children's understanding of nice and mean. Unpublished doctoral dissertation, University of Denver, Denver.

Harter, S. H. (1983). Developmental perspectives on the self system. In P. H. Mussen (Ed.), Carmichael's manual of child psychology (4th. ed.). New York: Wiley.

Knight, C. C. (1982). Hierarchical relationships among components of reading abilities of beginning readers. Unpublished doctoral dissertation, Arizona State University, Tempe.

Knight, C. C., & Nelsen, E. A. (1982, August). Hierarchical relationships among components of reading abilities in beginning readers. Paper presented at the Annual Meeting of the American Psychological Association, Washington, D. C.

Krus, D. J. (1976). Order analysis of binary data matrices. Los Angeles: Theta Press.

Krus, D. J., Bart, W. M., & Airasian, P. W. (1975). Ordering Theory and Methods. Los Angeles: Theta Press.

LaBerge, D., & Samuels, S. J. (1974). Toward a theory of automatic information processing in reading. Cognitive Psychology, 6, 293-323.

Papert, S. (1980). Mindstorms. New York: Basic Books.

Watt, D. (1983, October). Mindmirroring software. Popular Computing, pp. 65-68.

BEST COPY AVAILABLE

Recall of Verbal Spatial Information

In Graphic Form

Marie Knowlton

University of Minnesota

ABSTRACT

Recall of spatial material was required of 20 subjects who participated in a verbal computer game in which they explored a three-dimensional maze. After playing ADVENTURE for 15 minutes subjects were required to draw a map of the maze they had explored. The maps were rated for correctly positioning four separate locations and for total consistency in the spatial layout of the maps. Graphic representation of the spatial relationships was at the level of chance. Subjects did not follow conventional techniques of recording spatial material. This may be due to the piecemeal format of obtaining the material via a computer game, or to lack of redundancy in the information available from successive locations.

The design of successful computer based instructional materials necessitates the inclusion of all necessary and sufficient information for the required task. If the task is to be completed with optimal efficiency there must also be a certain level of redundancy which permits the interaction of different informational components. In order to explore the comprehension of textual materials in a more systematic manner Bieger (1980) proposed a taxonomy of categories of information in picture/text procedural materials. Three of these proposed categories (operational, spatial, and contextual) provided the necessary and sufficient information for experimental assembly tasks. Further exploration of one of these categories (operational information) led Knowlton (1983) to propose that functional uses of objects are accessed from their identities rather than from specific features which fulfull the procedure.

Method

To continue this exploration of picture/text informational content I will present an experiment examining the category of spatial information. Spatial information is more easily portrayed in graphic form (pictures) and it is hypothesized that since the information is readily portrayed graphically it is probably stored in memory in this manner rather than converted to a verbal representation. It would then follow that if the information is stored in a graphic form the recall of this information using a picture recall would be a more direct method of assessing learning of spatial relationships. To test this hypothesis I will use a design where directional information is presented by the subject in a verbal mode and test for recall of spatial configurations by a graphic mode.

The computer game ADVENTURE was used for this study. The objective of this game is to explore an underground, three-dimensional maze or cave. The format is completely textual. The player is provided with a verbal description of a location. S/he then directs the computer by entering a directional command (NORTH, SOUTH, UP, DOWN, etc.). The computer then provides a description of the new location in the relevant direction, and additional incidental information. Spatial relationships are fairly accurately represented according to a three dimensional model. That is, if a command of NORTH is entered on the first turn and SOUTH on the next turn the player is returned to the original location. Similarly, the consecutive commands UP and DOWN will return the player to the original location.

The subjects were 20 undergraduate students who were unfamiliar with this particular computer game or games similar to it. They were individually instructed in how to play the game, directed through the program to the entrance of the cave and then permitted to play without interference for fifteen minutes. The movements and commands were monitored and recorded. At the end of this time the computer screen was cleared and they were asked to draw a map of the section of the cave that they had explored. The maps were evaluated on correct placement and relative positions of four locations identified by a building, the cave entrance, a bird cage and a debris filled room. The cave entrance is south of the building. The bird cage is west of the entrance, and the debris filled room is west of the bird cage. A map consistent with these relationships is presented in Figure 1. (see Figure 1)

Each map was given an additional rating according to whether or not conventional procedures of map design were observed. That is, orienting the map with north to the top of the page and east to the right of the page. These directional conventions had to be maintained for the complete map. I must point out that some subjects indicated the same compass direction with two different spatial directions on the same map.

Results

The number and percent of correctly mapped spatial relationships for each of the pairs of locations is reported in Table 1 as well as the total number of correctly mapped relationships. Individual pairs of locations were correctly related 45 to 65 percent of the time. In addition, 12 of the 20 subjects observed conventional map coordinated with north to the top of the page and east located to the right. Seven subjects demonstrated east/west reversals or indicated the same compass direction in more than one direction. (see Table 1)

Discussion

This study investigates the encoding of verbal spatial information. The testing and evaluation of the verbal material was done by having subjects draw maps in order to avoid testing in a verbal mode since it was believed that spatial information would be easier to retrieve in a graphic form. The results strongly indicate that subjects are neither accurate nor conventional in the manner in which they choose to express their knowledge of spatial relationships. One subject who was completely unable to draw a map of any sort and finally after several attempts resorted to a written verbal description which was fairly accurate.

Of interest is the fact that less than half the subjects understood the relationship between the building and the entrance to the cave. Since the examiner was working directly with the subject during this part of the study it may be that attention was divided on trying to learn the game and the actual directional commands given to the computer. Another possibility is that this is a situation analogous to visually guided behavior. If the subject is involved in his/her own "locomotion" in space there is a more accurate/intentional encoding and recall of the operation. Certainly, the improved performance for the next two subsequent locations would support this suggestion.

It is also interesting that subjects do not necessarily follow conventional techniques of recording spatial material. East/west reversals occurred for three subjects and a North/south reversal for one subject. Inconsistencies in spatial layout such as having two different directions labeled west are not obvious to the subject drawing the map and probably indicate the piecemal organization of the cognitive representation. Subjects are encoding pairs of locations with little or no overall sense of spatial organization.

Further investigation is needed to understand more fully the encoding of spatial information. However, the implications of this study suggest that encoding of verbal spatial directions is not a consistent process. There may be a large component related to a subject's ability in mental imagery and familiarity with conventions of displaying spatial information. This cannot be determined by this study.

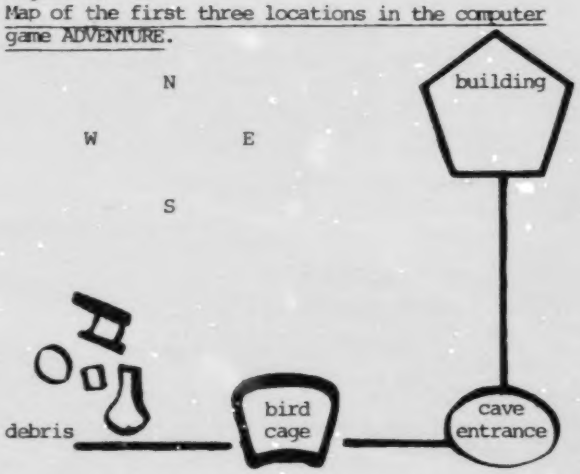
The lack of redundancy between computer and subject input may also effect performance. In this study the computer program supplies only information on locations. The subject supplies the directional information that provides the spatial relationships between two locations. Hence, verification of the spatial relationships between two adjacent location has to be inferred by the subject. Indirect feedback can come from the computer if the subject retraces routes previously traveled. When subjects did this, however, they were usually very disoriented or "lost" in the maze.

Finally, verbal compass directions are not a natural or reliable mode for inferring spatial relationships between locations for most people. Subjects performed almost at chance level on this task and were unaware of inconsistencies in their spatial organization. It remains to be investigated whether this is due to their inability to express spatial relationships graphically or whether they did not consider consistency a necessary part of information encoding.

Table 1.
Number and perdentage of correctly mapped relationships between adjacent locations.

spatial relationship	number correct	percent correct
1) house / cave	9	45
2) cave / bird cage	13	65
3) bird cage/debris room	10	50
total	32	53.3

Figure 1.
Map of the first three locations in the computer game ADVENTURE.



References

Bieger, G. & Glock, M. (1980). The informational content of picture-text assembly instructions, Technical Report No. 5. Ithaca: Cornell, Department of Education.

Knowlton, M., Keil, F., & Glock, M. (1983). The perception of objects and their functional uses. Technical Report No. 8. Ithaca: Cornell University, Department of Education.

INSTRUCTIONAL GRAPHICS IN COMMERCIAL COURSEWARE

Kathryn Lutz Alesandrini

The University of Iowa

Abstract

The effectiveness of computer courseware may lie in part with the nature of the graphics that are used. Little is known, however, about the extent to which commercial courseware incorporates instructional graphics and whether or not graphics are used to facilitate learning. The paper discusses relevant literature on instructional graphics according to three main categories of graphics including representational, analogical, and arbitrary. An informal survey of the use of instructional graphics in commercial courseware revealed some shortcomings. Although CBI can theoretically make use of interactive graphics, not all lessons surveyed did so. In math and science, about half of the surveyed lessons presented visual feedback after learner responses. Less than 1/3 of the lessons in social studies and language arts used any type of visual feedback. Of greater concern is the general lack of visual presentation of information in CBI despite growing evidence that pictures aid learning. Of the 60 CBI lessons surveyed, more than half used only a few graphics or included no graphics. Not one of the surveyed lessons used "analogical" graphics, a type of picture that may help a learner relate new or difficult information to more familiar concepts. Suggestions are made for the design and use of graphics in computer-based instruction.

Introduction

Instructional graphics or pictures may be classified on the basis of how they convey meaning including representational, analogical, or arbitrary (Gropper, 1963; Knowlton, 1966). Some previous reviewers of picture effects have dealt with only the first category--that is, graphics that are isomorphic with the objects or concepts that they represent (Alesandrini,1982a; Levin & Lesgold, 1978). Other reviews have also considered arbitrary or nonrepresentational graphics such as flowcharts and graphs (Alesandrini, in press; Levie & Lentz, 1982; Macdonald-Ross, 1977a). This paper briefly discusses research on all three types of graphics and considers how each type may play a crucial, yet different, role in the learning process. The use of each type in CBI is also surveyed and discussed. The terms pictures and graphics are used interchangeably here.

Representational Graphics

Representational graphics are those that share a physical resemblance with the thing or concept that the picture stands for. Use of representational pictures is supported by the research and theory on the potency of visual memory (Yuille & Marschark, 1983) and the importance of providing examples when teaching concepts (Tennyson & Park, 1980). Knowlton (1966) used the term realistic to describe this category of picture.

Representational pictures can either portray information directly or indirectly. Tangible objects or concepts may be represented directly. Although abstract concepts cannot be directly portrayed, they may be portrayed indirectly by showing their effects, results, instances, or exemplars (Gropper, 1963). An example is to portray "states of matter" by showing a candle that is lit because it

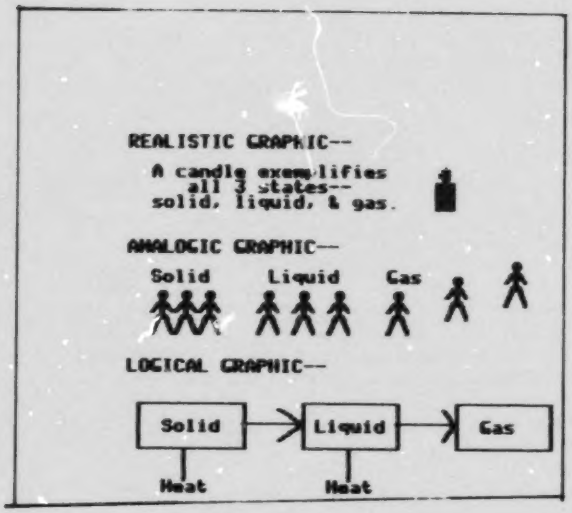


Figure One. Examples of the three types of pictures to illustrate the topic "states of matter" (Alesandrini, 1984).

exemplifies all three states. Another approach to illustrating abstract concepts representationally is to portray a concrete associate of the concept (Jones, 1983). Other have made up pseudo-examples that portray what the educator thinks the concept might look like if it were tangible (Cantu & Herron, 1978).

Most studies on pictures have tested the effect of supplementing verbal material with representational or realistic pictures. Dwyer (1978; 1982-83) and his associates have conducted over 100 studies involving more than 40,000 adult learners since 1965 to investigate the effects of realistic pictures that varied in the amount of detail from highly detailed colored photos to simple line drawings. The results of that research effort have generally shown that this type of picture facilitates learning. Findings by other researchers have generally paralleled the results of Dwyer, showing that representational pictures facilitate learning.

Representational graphics in computer-based instruction have a similar effect to those presented via other media. Simple line drawings presented via computer-based instruction facilitated science learning by college students compared to a read-only control condition (Rigney & Lutz, 1976). However, the graphics did not produce a facilitative effect compared to the verbal strategy of answering inserted questions (Alesandrini & Rigney, 1981). A related study investigated graphics that the learner could manipulate via a touch-sensitive screen (Alesandrini & Rigney, 1981). College learners in that study either reread the lesson content or manipulated computer-based graphics that were designed to relate specific concepts to the bigger picture. Rearranging and manipulating the graphic representation of the content facilitated performance on a picture recognition test but not on a verbal, multiple-choice test.

In sum, there is some evidence that representational graphics, whether they are presented via CBI or not, may be effective in facilitating learning.

When graphics are used to visually portray information in CBI, representational graphics are most commonly used according to the software surveyed in this study. In science, 71% of the surveyed packages included representational graphics (see Table 1). In the areas of math, language arts, and social studies, fewer lessons used this category of graphics with 40%, 25%, and 14%, respectively. For all areas, the percentages are higher for representational graphics than for the other two categories of graphics as discussed later in this paper.

Analogical Graphics

Analogical graphics convey a concept or topic by showing something else and implying a similarity. For this type of picture to be successful, the learner must recognize or be able to comprehend the object used as the analogy. According to learning theorists, new information will be better remembered if the learner relates it to prior knowledge (Reigeluth, 1983; Wittrock, 1974; 1977). Analogical pictures should help the learner interpret new information in light of prior knowledge and thereby facilitate learning. Figure 1 gives an example of an analogical graphic for the topic "states of matter." It implies that solids are like a closely grouped line of people, liquids like a looser line, and gases like an unassociated group of people.

Research evidence indicates that visual analogies can facilitate learning. Several studies found that learners can transfer information gained from an analogy to understand subsequent abstract information (Royer & Cable, 1975; 1976). Furthermore, analogical diagrams have been used to facilitate both learning (Mayer, 1975) and problem solving (Gick & Holyoak, 1983).

Some evidence suggests that it may not be necessary for learners to be

familiar with the analogy if an easily comprehended picture of the analogical object is presented. In one study, college students who read "The Mat Maker" chapter from Melville's Moby Dick with an analogical picture made fewer errors on a posttest (Davidson, 1976). No studies on the effects of visual analogies presented via CBI could be found.

Of the 60 CBI packages surveyed, none made use of analogical graphics. This observation should raise concern that current software packages may not be incorporating the most effective instructional strategies in their design.

Arbitrary Graphics

Another category of picture has been termed arbitrary or logical because these highly schematized visuals do not look like the things they represent but are related logically or conceptually. Knowlton (1966) referred to this type of picture as "logical," describing it as isomorphic with the "pattern and/or order of connection" of the represented state of affairs rather than isomorphic with the elements portrayed. Arbitrary pictures include graphs, flowcharts, networks, maps, tree diagrams, and other schematized charts and diagrams. An example of an arbitrary graphic is shown in Figure 1, consisting of a flowchart that schematically portrays the relationship between the three states of matter.

Arbitrary graphics have been used to structure and organize the information in text, taking a variety of specific forms such as structured overviews, networks, structural outlines, hierarchical mapping, and tree diagrams to name a few. Research indicates that most of these forms of graphic organizers of text facilitate learning. A structured overview graphically presents the key vocabulary from a new passage, showing the relationships between new concepts and familiar concepts (Barron, 1969). A meta-analysis of structured overviews found that graphic organizers do tend to facilitate learning (Moore & Readence, 1983). One recent study has confirmed earlier results and shown that structured overviews facilitate both lower- and upper-level comprehension (Snouffer & Thistlethwaite, 1983).

Learning has also been facilitated by structural outlines, consisting of hierarchical word trees with various levels of specificity, proceeding from the most general to the most specific concepts (Glynn & DiVesta, 1977). Another approach used has been termed conceptual hierarchy, where each node of the hierarchy represents one or more features that lower-level nodes have in common (Merrill & Tennyson, 1977). In several studies, showing a diagram of the conceptual hierarchy of the topic facilitated learning (Wilcox, Merrill, & Black, 1981; Wilson & Merrill, 1980).

There has been little research on the effects of schematized charts and diagrams on learning. In one study, showing learners an abstract diagram helped them solve a problem whose solution could be logically represented by the diagram (Gick & Holyoak, 1983). Interestingly, however, the diagram was only facilitative when learners were specifically instructed to use it in arriving at a solution.

The effects of showing learners charts or diagrams may vary with the learner's ability. In one study learners of low verbal ability learned more from a flow diagram that contained representational visuals (line drawings) while high ability learners did equally well with the abstract flow diagram and the more concrete version of the diagram containing line drawings (Holliday, Brunner, & Donais, 1977). In another study with young adults, learners with high verbal ability performed better on an identification task when critical attributes in the visual material were highlighted via arrows and they also made fewer errors when a two-dimensional chart was shown (Winn, 1981).

There has been some research on the use of flowcharts and other schematized visuals to represent directions and other algorithmic information. Most of the studies in this area have focused on how flowcharts facilitate the learner's speed in performing the visually-described procedures and also how they affect the number of errors made. Several studies have shown that flowcharts improved comprehension of directions as compared with directions given in a prose format (Kammann, 1975; Lewis, Horabin, & Gane, 1973; Wright & Reid, 1973). However, a more recent study found that flowcharts used to represent algorithmic information about lab procedures in a college chemistry course did not facilitate critical thinking, ability, time required to complete laboratory assignments, and final course grade (Coscarelli & Schwen, 1979).

Taken together, the studies on the effects of arbitrary pictures suggest that learning and performance may be facilitated by this type of visual information. As applied to learning new information, arbitrary graphics may serve

to abstractly portray the essence of new concepts and, thereby, facilitate learning.

Of the 60 CBI packages surveyed in this study, 13 used arbitrary pictures in presenting information (22%). Of those 13, nine lessons that used arbitrary graphics were in the area of mathematics. None of the lessons in social studies used this type of graphic. Yet many of the abstract concepts in the area of social studies and language arts could be made clearer and more memorable via flowcharts, schematic diagrams, graphs, etc. For example, the following flow diagram was used in a social studies lesson entitled "The Boston Tea Party" (Ohlhausen, 1982). The arbitrary graphic was built up step by step and then the "representatives" step was removed to visually explain the concept "taxation without representation."

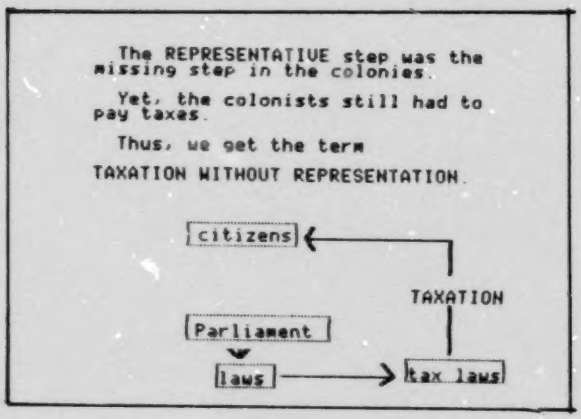


Figure Two. Arbitrary graphic used in social studies lesson.

Survey of Graphics in Courseware

Based on the evidence that all three types of pictures can facilitate learning, microcomputer courseware that portrays the main points visually should be more effective than equivalent all-verbal courseware or lessons that use visuals for decoration and motivation only. Abstract concepts and complex topics can be visually represented in a number of ways as explained by Gropper (1963). If visual portrayal in a representational or direct way is not possible, the content can be visualized via analogical graphics or arbitrary graphics. In order to determine the extent to which these three types of graphics are used in commercial courseware, an exploratory survey was conducted.

Sample

It was not economically feasible to obtain a systematically drawn sample of commercial courseware that was representative. Therefore, lessons that were available to the researcher were surveyed for this analysis. Many of the lessons are from the Minnesota Educational Computing Consortium (31 of the 60 surveyed).

Ratings

The 60 lessons were rated according to the amount of graphics contained in the lesson. Since total number of graphics contained in a lesson is confounded with the lesson's length, amount was not defined by number of graphics per se. Rather, the bases for categorization was the rater's judgement of the percentage of time that visual information was displayed on the screen relative to the length of the lesson. If graphic material was on the screen less than ¼ of the time, the lesson was rated as having few or no graphics. From ¼ to ½, the lesson was rated as having some graphics. The lesson was rated as having many graphics if visual material was on the screen more than half of the time. Two graduate students and the author each rated 20 lessons.

In addition to rating the amount of graphics in each lesson, the raters also scored each lesson for the presence or absence of each type of graphic--representational, analogical, and arbitrary--for presenting information during the lesson. No attempt was made to score the amount of the three types or their quality. All ratings (other than the general amount of graphics) were rated based on presence or absence of the target characteristic. For example, if the package used visual reward after a wrong answer, the presence of that feature was counted once for that lesson. Raters scored the presence or absence of the following six features in the 60 lessons: irrelevant graphics, representational graphics, analogical graphics, arbitrary graphics, visual feedback, and visual reward of wrong answers.

Results

Results of the ratings are given in Table 1 in the form of percentages. Although microcomputers have been touted as a new visual media with the potential for visual communication of information and interactive graphics (Alesandrini, 1982b;1983), most of the lessons surveyed used few or no graphics. Language arts and social studies were especially lacking in their use of graphics. Almost half of the science lesson used some graphics and a majority of the math lessons used

graphics. Overall, the general lack of graphics in the lessons was surprising and should concern educators. Happily, however, the use of irrelevant graphics was low as was the use of graphics to reward incorrect answers. Apparently, only a few lessons used graphics for motivation or decoration only and only 3 out of 60 lessons rewarded responses whether they were correct or incorrect.

None of the lessons used analogical graphics which runs contrary to the research and theory on the usefulness of this type of graphic. An example of the use of this type of graphic may be found in a lesson to teach educators about computer-based instruction (Alesandrini, 1984). The following analogy from that lesson makes the point that visual presentation of information can sometimes be more helpful and valuable than a verbal presentation.

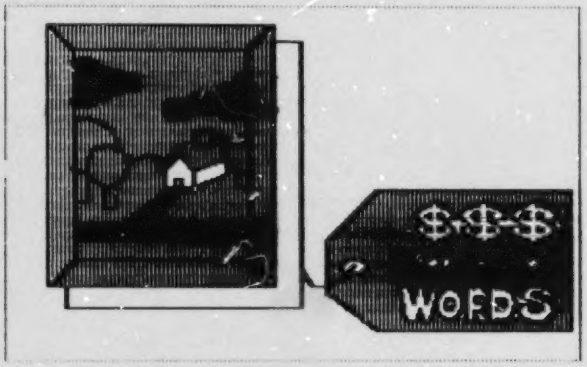


Figure Three. An analogical graphic for making a point about the value of visual information.

Representational graphics were by far the most commonly used type of graphic in the software surveyed, except in the math lessons were the incidence of lessons containing arbitrary graphics was one greater than the incidence of lessons containing representational graphics. The reason for this pattern is open to speculation. Perhaps lesson designers are not aware of the potential benefits of analogical and arbitrary graphics.

Computer-based instruction (CAI) has the potential of presenting graphics that a learner may interact with and manipulate, thereby learning from the consequences of those actions. The incidence of lessons using interactive graphics and visual feedback, however, was rather low. Only in the area of science did most lessons give feedback visually (often along with other types of feedback as well). The language and social studies areas were again very low in their use of this type of graphic.

Summary

In summary, a disappointingly low percentage of the 60 commercial lessons examined in this study used graphics to communicate the information being taught or drilled. When graphics were used, lessons tended to rely on representation or realistic graphics rather than analogical or arbitrary graphics despite evidence that the latter two types can facilitate learning. Fortunately, few of the surveyed lessons used graphics for decoration or to reward wrong answers. Based on these results, developers of computer-based instruction (CBI) and classroom educators using CBI may want to give more attention to graphics or a lesson and possibly supplement the lesson with analogical graphics or arbitrary graphics as needed.

(continued on next page)

Table 1

Graphics in Commercial Software

	Math	Science	Lang. Arts.	Social Studies
Packages Surveyed	20	14	12	14
(MECC)	8	8	5	10
Amount of Graphics				
Few/no Graphics	35%	57%	67%	86%
Some Graphics	25%	36%	25%	7%
Many Graphics	40%	7%	8%	7%
Irrelevant Graphics	15%	7%	25%	0%
Representational Graphics	40%	71%	25%	21%
Analogic Graphics	0	0	0	0
Arbitrary Graphics	45%	21%	8%	0
Visual Feedback	45%	57%	17%	29%
Visual Reward for Errors	5%	0	17%	0

References

Alesandrini, K. L. (1984). CBI: The Educators new tool. [software] Iowa City, IA: The University of Iowa. (Published by MicroConnect, Santa Monica, CA.)

Alesandrini, K.L. (1982). Imagery-eliciting strategies and meaningful learning. Journal of Mental Imagery, 6, 125-140.

Alesandrini, K. L., & Rigney, J. W. (1981). Pictorial practice and review strategies in science learning. Journal of Research in Science Teaching, 5, 465-474.

Barron, R. F. (1969). The use of vocabulary as an advance organizer. In H. L. Herber & P. L. Sanders (Eds.), Research in Reading in the Content Areas: First Year Report. Syracuse, New York: Syracuse University.

Cantu, L., & Herron, J. (1978). Concrete and formal Piagetian stages and science concept attainment. Journal of Research in Science Teaching, 15, 135-143.

Coscarelli, W. C., & Schwen, T. M. (1979). Effects of three algorithmic representations on critical thinking, laboratory efficiency, and final grade. Educational Communication and Technology Journal, 27, 58-64.

Davidson, R. E. (1976). The role of metaphor and analogy in learning. In J. R. Levin & V. L. Allen (Eds.), Cognitive learning in children: Theories and strategies (pp. 135-162). New York: Academic Press.

Dwyer, F. M. (1978). Strategies for Improving Visual Learning. State College, Pa: Learning Services.

Dwyer, F. M. (1982-83). The program of systematic evaluation--A brief review. International Journal of Instructional Media, 10, 23-38.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. Cognitive Psychology, 15, 1-38.

Glynn, S. M., & Di Vesta, R. J. (1977). Outline and hierarchical organization as aids for study and retrieval. Journal of Educational Psychology, 69, 89-95.

Gropper, G. L. (1963). Why is a picture worth a thousand words? AV Communication Review, 11, 75-79.

Holliday, W. G., Brunner, L. L., & Donais, E. L. (1977). Differential cognitive and affective responses to flow diagrams in science. Journal of Research in Science Teaching, 14, 129-138.

Jones, S. (1983). Stereotype in pictograms of abstract concepts. Ergonomics, 26, 605-611.

Kammann, R. (1975). The comprehensibility of printed instructions and the flowchart alternative. Human Factors, 17, 183-191.

Knowlton, J. (1966). On the definition of "picture". AV Communication Review, 14, 157-183.

Levie, W. H., & Lentz, R. (1982). Effects of text illustrations: A review of research. Educational Communication and Technology Journal, 30, 195-232.

Levin, J. R., & Lesgold, A. M. (1978). On pictures in prose. Educational Communication and Technology Journal, 26, 233-243.

Lewis, B. N., Horabin, I. S., & Gane, C. P. (1973). Case studies in the use of algorithms. New York: Pergamon Press.

Macdonald-Ross, M. (1977a). Graphics in text. In L. S. Shulman (Ed.), Review of research in education (No 5 pp. 49-85). Itasca, IL: Peacock.

Mayer, R. E. (1975). Different problem solving strategies established in learning computer programming with and without meaningful models. Journal of Educational Psychology, 67, 725-734.

Merrill, M. D., & Tennyson, R. D. (1977). Teaching concepts: An instructional design guide. Englewood Cliffs, NJ: Educational Technology.

Ohlhausen, M. (1982). The Boston Tea Party. [Software]. Iowa City, IA: The University of Iowa.

Rigney, J. W., & Lutz, K. A. (1976). Effect of graphic analogies of concepts in chemistry on learning and attitude. Journal of Educational Psychology, 68, 305-311.

Royer, J. M., & Cable, G. W. (1975). Facilitated learning in connected discourse. Journal of Educational Psychology, 67, 116-123.

Royer, J. M., & Cable, G. W. (1976). Illustrations, analogies, and facilitative transfer in prose learning. Journal of Educational Psychology, 68, 205-209.

Snouffer, N. K., & Thistlewaite, L. L. (1983). The effects of the structured overview and vocabulary preteaching upon comprehension levels of college freshmen reading physical science and history materials. Journal of the Association for the Study of Perception, Fall, 11-16.

Tennyson, R. D., & Park, O. C. (1980). The teaching of concepts: A review of instructional design research literature. Review of Educational Research, 50, 55-70.

Wilcox, W. C., Merrill, M. D., & Black, H. B. (1981). Effect of teaching a conceptual hierarchy on concept classification performance. Journal of Instructional Development, 5, 8-13.

Wilson, B. G., & Merrill, M. D. (1980, April). Effects of structural instruction and sequence in learning a conceptual hierarchy. Paper presented at the annual meeting of the American Educational Research Association, Boston, MA.

Winn, W. (1981). Effect of attribute highlighting and diagrammatic organization on identification and classification. Journal of Research in Science Teaching, 18, 23-31.

Wittrock, M. C. (1974). Learning as a generative process. Educational Psychologist, 11, 87-95.

Wittrock, M. C. (1977). Learning as a generative process. In M. C. Wittrock (Ed.), Learning and instruction (pp. 621-631). Berkeley: McCutcheon.

Wright, P., & Reid, F. (1973). Written information: Some alternatives to prose for expressing the outcomes of complex contingencies. Journal of Applied Psychology, 160-166.

Yuille, J. C., & Marshark, M. (1983). Imagery effects on memory: Theoretical interpretations. In A. A. Sheikh (Ed.), Imagery: Current theory, research, and application (pp. 131-155). New York: Wiley.

The Effects of Computerized Testing and Attitudinal Response of Preservice Early Childhood Education Majors

Jack V. Powell

L. David Weller

University of Georgia

ABSTRACT - The purpose of this study was to investigate the effects of taking tests via computer as compared with the more traditional paper-pencil method. Subjects' attitudes toward contents of a prerequisite course to student teaching were also assessed by using a semantic differential type instrument. Relationship between achievement and SAT scores was also examined. Dependent T-tests results showed gaines in achievement for both groups. A significant difference between the groups on a Planning subtest and on the overall semantic differential for the experimental group were also found. Correlation coefficients showed no significant relationship between achievement and SAT scores.

With respect to numerous advancements in information technology Gleason (1981) has predicted that by 1985, one million microcomputers will be in use on the elementary and secondary school levels within the United States. These advancements are also reflected through a delineation of ten ways in which information technology can be used to improve education and the historical innovative applications of the potential of computers within the department of elementary education at the University of Georgia (Licklider, 1981; Powell, 1981). It, therefore, seems apparent that colleges of teacher education have an obvious and quite demanding challenge to meet for years to come.

As a means of addressing this challenge, countless recommendations have been offered to educators through statements of computer literacy related to hardware, software, programming, various applications within subject fields, and suggestions for determining affective aspects of human-machine interactions. For example, Johnson, Anderson, Hansen and Klassen (1980) presented a revised list of objectives for computer literacy within the cognitive and affective domains as a result of feedback from experts in fields affiliated with industry, education and numerous professional societies. Similarly, Dearborn (1983) reported that computer literacy in Alexandria, Virginia Schools is now parallel to the three R's in importance, and he explicitly outlines a conceptualization of computer literacy under six topical areas. These include: computer operations; computer literacy strands; how to incorporate computers into the curriculum; educational applications of the microcomputer; computer programs; and the development of problem solving/logical thinking skills. Downing (1982) also focused attention on the computer literacy issue when he suggested that programs that are intended to develop literacy usually include introductory topics to keyboard, functions of various components of the computer, programming languages, applications as well as historical and societal impact of computers.

Amid the body of existing literature on computer literacy, resulting from the computer revolution, there exist many questions about what roles teacher education departments within universities should play in order to ensure that its graduates will become prepared to assume positions in elementary or secondary classrooms where many of the students are successful programmers and home-computer users (Center for Advanced Technology, 1983). Accompanying the rapid advancements in computer technology, and the logistics of developing computer literacy (or competence) in classroom teachers,

educational institutions are presented with a challenge and a dilemma as well. In addition to perpetuating the traditional teacher education curricula, there is also the problem of reaching consensus for the development of courses solely devoted to computer applications. According to Sherwood, Connor, and Goldberg (1981), the problem is not related to potential enrollment, but an administrative decision as to whether or not to include such course as an integral part of elementary preservice curricula. Judd (1983) also addressed the need for prospective teachers to acquire computer literacy when he stated that "a teacher who won't have a computer in the classroom is like a ditch digger who won't learn to use a steam shovel" (p. 120).

The question now is what is computer literacy and how may it be operationalized within colleges of teacher education? Lee (1982) defines "computing literacy as a combination of skills emboided in applications software that allow you to drive down computing streets safely and confidently" (p. 60). He further commented that the majority of one's practical knowledge will come through observation and, more importantly, through personal use with a computer either individually or in a group setting. The stages of this competency, as Lee defined them, are "using the computer as machines, as a clerk doing data entry does; using them as tools, as a manager doing financial analysis does; and using them as creative instruments, as a writer or a programer does" (p. 60). On the other hand, Lipkin (1983) defined computer literacy, in broad terminology, as "the ability to use the computer as an aid to problem solving in all spheres of human activity, as appropriate" (p. 26). In addition, Morsund (in Battista and Krockover (1982) stated that the topic under discussion may be defined "as the knowledge of the non-technical and low-technical aspects of the capabilities and limitations of computers, and of the social, vocational and educational implications of computers" (p. 1). However, considering the existence of computer assisted instruction in the schools and the various programming languages (e.g., LOGO, BASIC and PILOT), some knowledge of programming among other literacy objectives for teacher educators cannot be overlooked in preparing for in-service employment where computer applications have gained prominence. It has been additionally documented by Grady and Poirot (1983) that "all teachers should attain a certain level of computer literacy" (p. 79). They suggested that the following topics should encompass needed competencies for teachers:

1. computer systems
2. computer terms, jargon, and colloquialisms
3. educational, albeit instructional uses of computers
4. history of computers and calculating
5. problem-solving techniques; computers and problem solving
6. nonschool uses of computers and
7. relationships between society and computers (p. 79)

Within the scope of developing some aspects of computer literacy in preservice teachers, this research paper focused on the experimental results of an instructional use of a time-sharing computer, namely interactive testing. Answers to the following questions were therefore sought: Were there significant differences on pre-and post-test achievement relating to course content between experimental and control groups? Were there any attitudinal differences between the two groups of students as they relate to the topics within the course content? How does student academic performance on course content correlate with their SAT scores? The variable sex was not considered for analysis since only two of the 65 students were males.

METHOD - Subjects for this study were 65 undergraduate students majoring in Early Childhood Education at the University of Georgia during Winter and Spring Quarters 1983. For each quarter respectively, 32 and 33 preservice beginning teachers were enrolled in a prerequisite course to student teaching entitled Program and Curriculum Development. For this investigation, two groups were randomly assigned: one for testing by interactive time-sharing computing and the other for testing by paper-pencil means. On the first day of each quarter, both groups experienced a drill and practiced exercise via computer, in order to nullify the Hawthorn effect. This activity was related to teaching arithmetic in the elementary school. Subsequently, a Semantic Differential Instrument (SDI) relating to the outline of the course, and developed by the second author was administered (Weller, 1980). On the second day of each quarter, Group One took a pretest via computer in a time-sharing computer laboratory located in the College of Education, and Group Two was pretested in a regular classroom setting. On the last day of class for the quarters, the groups again responded to the SDI in

a large group setting. Finally, during the regular examination periods, post-testing was administered to each group in the modes described above. SAT scores which were determined to be incorporated into the design of this study were gathered.

INSTRUMENTS - A fifty-item multiple choice pretest and posttest (hereafter referred to as Achievement Test) were developed by the first author two quarters in advance of this investigation. After being administered by paper-pencil to other students enrolled in the same course prior to the beginning of this investigation, both tests were determined to have construct validity as reflected by predetermined objectives for the course. A BASIC (Beginner's All-Purpose Symbolic Instruction Code) computer program was also written by the first author for computerized testing. Fifty points were the maximum score for each test.

The SDI was adapted to the sequential topical outline of the course - Planning, Teaching, Classroom Management, Code of Ethics, Tests and Integrative Computer Activity. Six sets of bi-polar adjectives were included under each heading. This instrument was used to assess subject attitudes toward the scope of the course.

RESULTS - Selected programs from the Statistical Analysis System (SAS) were used to analyze student test results.

The Dependent T-test was used to determine if there were any significant differences between the two groups of students on achievement regarding pre-and post-test results as they related to course content. While both experimental and control groups showed gains, no statistically significant differences were found between the two groups.

Pre-and post-test results on the SDI, as analyzed by the Dependent T-test, yeilded a significant difference between experimental and control students in their attitude toward the Planning subtest of the instrument. A statistically significant difference was also detected on the overall instrument itself by experimental students (Table 1).

To determine if there was a significant difference between students' pre-and post-test results assessing gains in course content and their SAT scores, the Pearson Correlation Coefficient was applied to student data (Table 2). Although both groups experienced significant gains on the achievement test as shown in Table 1, results did not reveal a significant relationship between SAT scores and total achievement in class.

(INSERT TABLE 1 HERE)

(INSERT TABLE 2 HERE)

DISCUSSION - The findings reported in this study suggest that students can achieve equally as well whether pre or post tests are given via computer or traditional means which could be attributed to the effectiveness of instruction. On the other hand, attitudinal differences between the groups toward instructional planning and the overall scope and sequence of activities within the course could be related to course emphasis on considering computers in instructional planning and the mode of test administration. SAT scores, perhaps, cannot be used as a predictor of successful achievement in the course that teaches the aspects of program and curriculum development.

Regarding the results on SDI for both groups, an unexpected ceiling effect was observed in the plot of pre-post test results on the Planning, Classroom Management, and the Code of Ethics subtests which seemingly, is due to the high regard for these categories which might have been acquired in previous methods courses. If more scores had been at or near the middle range of these subtests, noticeable pre-and post-test results probably would have been revealed. For future research, a revision in the structure of the SDI to make it a "forced choice" instrument by adding such words as "more" or "most", "less" or "least" before some of the bi-polar adjectives might yield different results. A soloman Four-Group Design (Campbell & Stanley, 1963) could perhaps reveal results beyond the confines of this investigation.

Table 1 - Dependent T-Test Results for Achievement, Six Subtest of SDI and Total SDI

Varible	T Value		T Prob.	
	Comp. Gp.	Trad. Gp.	Comp. Gp.	Trad. Gp.
Total Achievement	-4.07	-6.25	.0003	.0001
Planning	-2.39	-1.11	.0227	.2739
Teaching	-1.18	-1.06	.2449	.2992
Classroom Management	-0.65	-1.02	.5192	.3141
Code of Ethics	-1.60	-0.44	.1199	.6652
Tests	-1.01	-0.92	.3188	.3644
Computers	-0.99	-0.94	.3315	.3569
Total SDI	-1.92	-0.68	.0534	.4992

Table 2 - Correlation Coefficients of SAT Scores by Achievement Scores*

	SATV	SATM	SAT
Preachievement	0.12	0.16	0.16
Postachievement	-0.17	-0.08	-0.13
Total Achievement	0.19	0.19	0.21

*No correlations sig., HO: RHO = 0 ($p < .01$)

REFERENCES

1. Battista, Michael T., & Krockover, Gerald H. The development of the computer literacy of preservice elementary teachers: an empirical model. Paper presented at the Association of Teacher Educators National Conference, Orlando, Fla., 1982.

2. Campbell, Donald T. and Stanley, Julian C. Experimental and Quasi-Experimental Designs for Research on Teaching. In Gage, N.L. (ed.) Handbook of Research on Teaching, Rand McNally & Co., 1963.

3. Center for Advanced Technology in Education. What is computer literacy? Advance, Fall 1983, College of Education, University of Oregon, Eugene Oregon.

4. Dearborn, Donald E. Computer literacy. Educational Leadership, 1983, 41, 32-34.

5. Downing, Bruce. More on planning for computers in education. Commodore Magazine, 1982, June/July, 27-28.

6. Gleason, G. T. Microcomputers in education: The State of the Art. Educational Technology, 1981, 21, 7-18.

7. Grady, M. Tim & Poirot, James L. Teacher comptence: What is needed. Computers in curriculum and Instruction. Alexandria, VA: Association for Supervision and Curriculum Development, 1983.

8. Johnson, David C., Anderson, Ronald E., Hansen, Thomas, P., & Klassen, Daniel L. Computer literacy - what is it? The Mathematics Teacher, 1980, 73, 91-96.

9. Judd, Wallace. A teacher's place in the computer curriculum. Phi Delta Kappan, 1983, 65, 120-122.

10. Lee, The. Squaring off over computer literacy. Personal Computing, September 1982, 58 . . . 72.

11. Licklider, J. C. R. Impact of information technology on education in science and technology. The Journal of Computers in Mathematics and Science Teaching, Fall, 1981, 21-26.

12. Lipkin, John P. Equity in computer education. Educational Leadership, 1983, 41, 26.

13. Powell, Jack V. A position paper on computer applications to cognitive assessment and advisement in elementary teacher education. Paper presented at the Ohio State University Conference, Fresh Perspectives on the Improvement of Teacher Education, Columbus, Ohio, 1981.

14. Sherwood, Robert D., Connor, James V., Goldberg, Kenneth P. Developing computer literacy and competency for preservice and inservice teachers. The Journal of Computers in Mathematics and Science Teaching, 1981, 1, 23-24.

15. Weller, L. D. "Project eureka: Teachers of the gifted and talented." Title IV, Part C grant, Region 10 Education Service Center, Richardson, Texas, 1980.

STRATEGIES FOR DESIGNING DRILL AND PRACTICE PROGRAMS FOR COMPUTERS

DAVID F. SALISBURY

FLORIDA STATE UNIVERSITY

Abstract

This paper addresses two questions: "What is the role of drill and practice in learning?" and, "How should drill and practice programs for computers be structured so as to fulfill this role most effectively?" The paper attempts to answer the first question by presenting a summary of recent research in modern cognitive theory related to "automaticity of subskills". To answer the second question, the paper evaluates the strengths and weaknesses of three computer drill and practice strategies in light of modern cognitive theory and then describes a prototype strategy which exhibits most of the positive characteristics of the previous three strategies.

Strategies for Designing Drill and Practice Programs for Computers

This paper addresses two questions: "What is the role of drill and practice in learning?" and, "How should drill and practice programs for computers be structured so as to fulfill this role most effectively?" The paper attempts to answer the first question by presenting a summary of recent research in modern cognitive theory related to "automaticity of subskills". To answer the second question, the paper evaluates the strengths and weaknesses of three computer drill and practice strategies in light of modern cognitive theory and then describes a prototype strategy which exhibits most of the positive characteristics of the previous three strategies.

The Role of Drill and Practice in Learning--Automaticity of Subskills

Many of the modern theories related to how learners perform complex tasks such as reading and computing suggest that in order for a learner to be able to efficiently perform these complex learning tasks, performance of the lower level component skills must be automatic (Anderson, 1980; Gagne, 1982). This is generally referred to as "automaticity of subskills" and is based on the observation that as a skill becomes more practiced and automatic it requires less and less attention and interferes less with other ongoing cognitive processes (Spelke, Hirst, & Neisser, 1976). Consider the musician learning a new piece of music. Once the mechanics of the piece have been mastered, the musician can then focus attention on interpretation. The implication of this research is that the role of practice and drill in learning is to bring the learner to the level of "automaticity" on lower level component skills so that the learner can more readily perform some higher level complex skill.

The purpose of this paper is to describe a highly sophisticated drill and practice strategy for computers designed to bring a skill to the level of "automaticity" as quicky and as efficiently as possible. In order that the reader may appreciate the advantages of this drill and practice strategy we will first describe three less sophisticated drill and practice strategies examining their respective weaknesses and strengths.

Strategy One

Probably the most simplistic drill and practice strategy is that which is commonly employed by a tutor or teacher using flashcards. Typically, a stimulus is written on one side of the card and is presented to the learner. After the learner responds, reinforcement or corrective feedback is given. The card is then placed at the end of the stack of cards and the next card in the stack is presented to the learner. The learner simply keeps going through the stack of flashcards in sequence until all of the items have been learned.

There are several reasons why this drill and practice strategy is not very effective and should definitely be avoided. First of all, the learner must go through all of the items before seeing an item for the second time. It is obvious that the difficulty of learning the correct response to each item will be determined by the number of total items in the pool. In other words, if the item pool consists of only a few items, learning them will be fairly easy as compared to the case of a pool containing a very large number of items. The more items there are to be learned, the greater the chances for confusion. This is typically referred to as interference (Gagne, 1977; Hilgard & Bower, 1975). The amount of interference increases as the number of items in the pool increase.

Secondly, this strategy is highly inefficient due to the fact that an equal amount of time will necessarily be spent on both the learned and the unlearned items. In other words, the learner is required to respond to all items each time through the list even if some of the items have already been learned. There is no provision for the learner to concentrate effort on the unlearned items. A third problem with this strategy is that the items are always presented to the learner in the same sequence. This sets up a "serial learning effect" which allows the learner to use the item sequence as a context cue for responding to an item. The learner may not be able to respond correctly to the item when the sequence is altered.

Strategy Two

A somewhat more sophisticated strategy is one in which the items are separated into an "item pool" and a "review pool". This type of strategy is sometimes used by tutors or students using flashcards but is more commonly employed on computers. The "item pool" contains all of the items to be learned and the "review pool" contains items which have already been learned but need to be reviewed periodically. When an item has been learned it is moved into the "review pool" to be reviewed later. The criterion for moving an item into the "review pool" can be set so that the learner must respond correctly to that item a certain number of times or, even better, a certain number of times in a row.

In the first strategy described, the learner simply went through the list of items to be learned again and again until all items had been learned. In Strategy Two, the computer (or tutor) records the number of times the item has been presented and the number of times the learner has responded correctly to that item. The number of times the learner has responded correctly to an item is set back to zero whenever the learner responds incorrectly to that item.

This strategy is somewhat more efficient than the first one described due to the fact that unnecessary time is not spent on learned items because learned items are moved into the review pool. However, the problem of interference is still present for large item pools.

Strategy Three

To create a drill and practice strategy which does reduce the interference problem requires a drill program which contains a "working pool", an "item pool", and a "review pool". This type of drill paradigm is much too complicated to be attempted by a tutor or student using flashcards and requires the management and record keeping

---

Insert Figure 1 about here

---

capabilities of a computer. As shown in Figure 1, the "item pool" contains all of the items to be learned. The "working pool" contains only a few items--items which the learner is actually working on at the present time. The size of this "working pool" might be seven plus or minus two items (Miller, 1956) depending on the difficulty of the items and the age level of the audience of learners. When an item in the "working pool" has been learned, that item is moved into the "review pool". The "working pool" is refilled by first checking to see whether

or not any items in the "review pool" need to be reviewed. If so, these reveiw items fill the vacancies in the "working pool". If no items need to be reviewed, a new item is brought into the "working pool" from the "item pool". Each time the items in the "working pool" have been displayed the items are shuffled and presented the next time in a different sequence.

This approach is similar to the idea of "progressive part" learning suggested by McGeoch and Irion (1952). This involves review of a few old items as new ones are introduced. Many of the computer drill and practice programs being designed today do not provide for the maintaining of a record which keeps track of items which are still to be learned and items which have been learned but need to be reviewed by a particular learner and which would allow a learner to end a practice session at any point one day and return the next day picking up where he or she left off--working on the particular items she was working on during the last session. In the literature there is much evidence that short, spaced periods of practice give better results than concentrated practice periods (Anderson, 1980; Gay, 1973). Therefore, a drill and practice strategy should include provision for this type of record keeping and review.

The selection of items to enter into the "working pool" from the "item pool" can be done in a number of ways on the computer. One way is to select the items from the item pool in sequence. This may be the best approach if there is some particular relationship among the items which suggests that a prescribed sequence for introducing the items to the learner would be desirable. For example, when multiple discriminations are being learned, it is often advantageous to introduce two stimuli which are very similar together and to emphasize the distinctive differences between the two (Gagne, 1977, p. 108).

Items could also be selected from the "item pool" on a random basis. This would generally be the preferred method in order to eliminate the use of the sequence as a context clue. In the case of items involving numbers, the computer can generate the items using "item forms". Parameters for the values to appear in particular items can be set so that items involving values no larger than, say ten, will be generated.

The drill and practice strategy described in Figure 1, although fairly sophisticated and much better than the previous two strategies, still has a couple of problems which make it less efficient than it might be. First of all, if there is any possibility that the learner may already know some of the items to be learned, then it would be useful to provide some pretesting mechanism. All items could be pretested during the first session or each item could be pretested when it is first introduced. Items responded to correctly would be eliminated from the exercise.

Secondly, although this strategy makes provision for review, it is not " spaced review". In other words, the strategy could be made more efficient by providing three "review pools" rather than only one. These would most likely be a "daily review pool", a "weekly review pool", and a "monthly review pool". When an item is learned it would move from the "item pool" into the "daily review pool". This item would be brought back into the "item pool" when it has not been introduced for a day. Once the learner demonstrates the ability to respond correctly to this item even though it is only introduced once a day, the item would be moved into the "weekly review pool" and eventually into the "montly review pool". Ultimately, the item would drop out of the exercise completely. Completion of the drill would involve reviewong items on a monthly basis until all items are completely dropped from the exercise.

Strategy Four

Figure 2, adapted from Atkinson (1974) shows a drill and practice strategy for teaching a paired associate task. This stragegy is very

---

Insert Figure 2 about here

---

sophisticated in that it exibits all of the desirable characteristics of the previous strategies and also includes a pretest and "spaced review". The first three boxes (labeled 1,2,and 3) outline the procedure for filling the "working pool". The first step is to check to see if any review items are ready to be reviewed (box 3). If not, a new item from the "item pool" is used to fill the vacancy in the "working pool". Once the "working pool" is full, an item is selected at random from the pool and its state identified (box 4). As shown in boxes 6 through 11, the item may be presented to the learner in any one of

six states. If it is an item which has just entered the "working pool" for the first time, it will be presented in a "pretest" state (state 1, box 6). If the learner already knows the correct response to this item it will be deleted from the exercise, otherwise, its state will be changed to state 2 (box 7) and the next time the item appears it will be presented in "forward drill" state. "Forward drill" simply means that the first component of the paired associate will be presented to the learner and the learner is to respond by giving the second component of the pair. In "backward drill" state (state 3, box 8), the second part of the pair is presented to the learner and the learner furnishes the first part of the pair. For example, in learning the bones of the body, the student would be presented with a diagram of the body and asked to name a particular bone. This is "forward drill" state. In "backward drill" state, the learner would be given the name of a bone and asked to indicate its location on the diagram.

Once the learner has mastered the item in both "forward drill" and "backward drill" states, the state of the item is changed to state 4 (box 9) which is a "daily review" state. Hence, this item will not appear again until the next day. States 5 and 6 are "weekly review" and "monthly review" states respectively. Note that the criterion for moving an item from one state to the next can be varied. Research needs to be conducted to determine the most effective criterion levels for moving an item from one state to the next. Criterion for each item and each student can be varied dynamically based on individual and/or group performance.

Although the logic behind this strategy is fairly complex, when the exercise is presented on a computer is seems very simple and straightforward to the learner. The learner is unaware of all the management decisions and record keeping going on inside the computer. The computer is capable of utilizing this type of sophisticated logic in order to present practice items to the learner in an optimally efficient manner.

Summary

The computer provides capabilities heretofore unavailable for optimizing practice and drill for learners. There is no need to apologize for using the computer for drill and practice. Providing drill and practice efficiently is no less complex than list keeping or word processing. Research in cognitive learning theory is giving more credibility to the role of drill and practice in learning by suggesting that "automaticity of subskills" is an essential prerequisite for performance of higher cognitive skills. Computer drill and practice programs can contribute to the process of helping a skill become automatic by providing sophisticated drill paradigms. Research related to issues such as interference, serial learning effects, spaced practice, and spaced review should be considered in the design of computer drill and practice programs. The highly sophisticated drill and practice strategy described in this article exhibits desirable qualities needed in computer drill and practice programs.

References

Anderson, J. R. Cognitive psychology and its implications. San Francisco: Freeman, 1980.

Atkinson, R. C. Teaching children to read using a computer. American Psychologist, 1974, 29, 169-178.

Gagne, R. M. The conditions of learning (3rd ed.). New York: Holt, Rinehart, & Winston, 1977.

Gagne, R. M. Developments in learning psychology: implications for instructional design; and effects of computer technology on instructional design and development - an interview with Robert M. Gagne. Educational Technology, June, 1982, 11-15.

Gay, I. R. Temporal position of reviews and its effect on the retention of mathematical rules. Journal of Educational Psychology, 1973, 64, 171-182.

Hilgard, E. R., and Bower, G. H. Theories of learning (4th ed.). Englewood Cliffs: Prentice Hall, 1975.

McGeoch, J. A., and Irion, A. L. The psychology of human learning, (2nd ed.). New York: McKay, 1952.

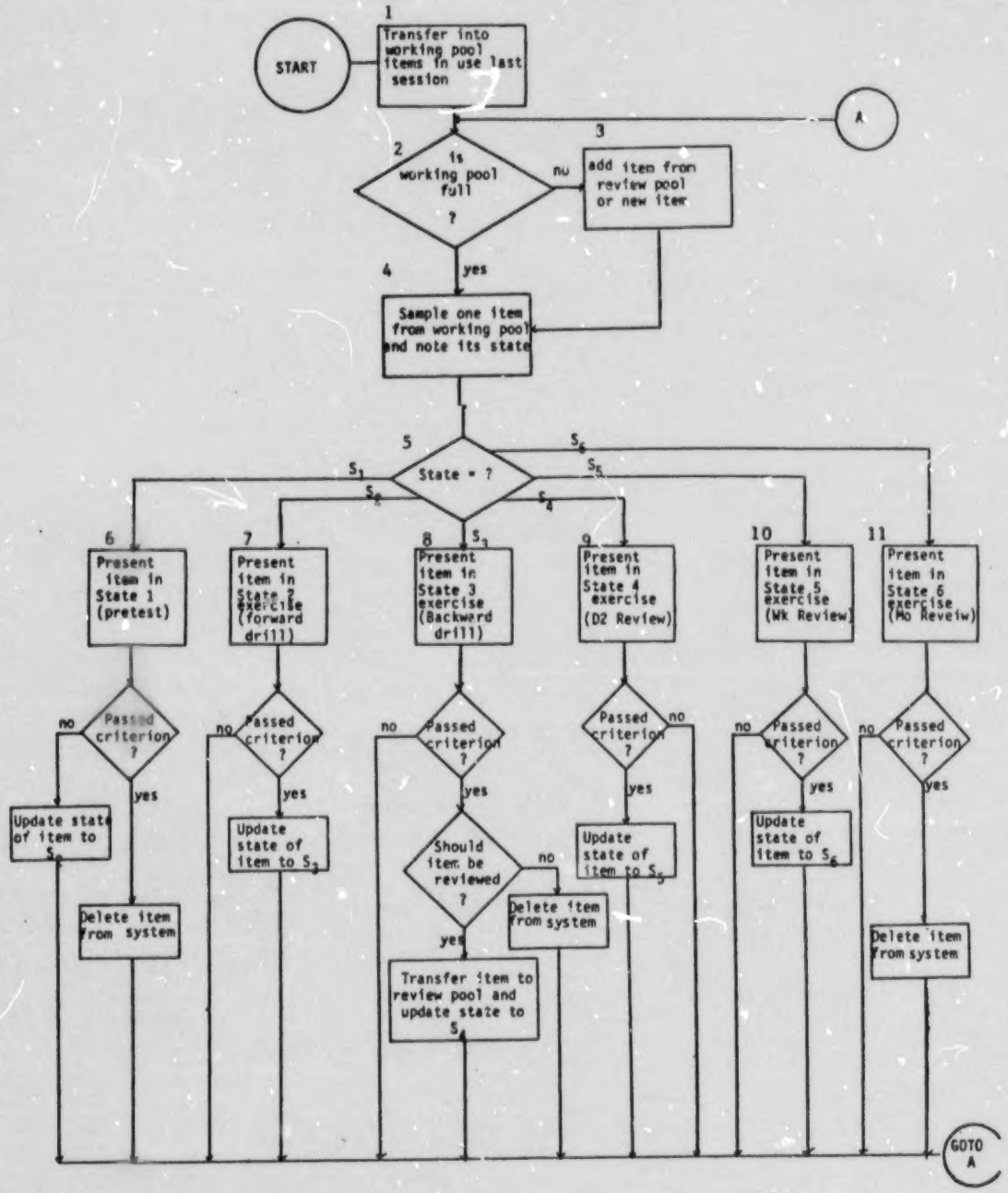
Miller, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review, 1956, 63, 81-97.

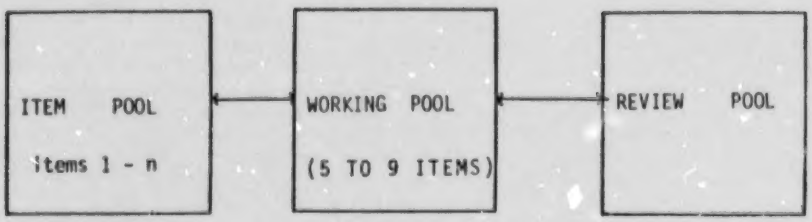
Spelke, E., Hirst, W., and Neisser U. Skills of divided attention. Cognition, 1976, 4, 215-230.

Figure Captions

Figure 1. Structure of a Drill and Practice Strategy Containing Three Pools.

Figure 2. Flowchart for a Sophisticated Computer Drill and Practice Strategy.


START
1
Transfer into working pool items in use last session
A
2
Is working pool full ?
no
3
add item from review pool or new item
4
yes
Sample one item from working pool and note its state
5
State = ?
S_1
S_2
S_3
S_4
S_5
S_6
6
Present item in State 1 (pretest)
7
Present item in State 2 exercise (forward drill)
8
Present item in State 3 exercise (Backwerd drill)
9
Present item in State 4 exercise (D2 Review)
10
Present item in State 5 exercise (Wk Review)
11
Present item in State 6 exercise (Mo Reveiw)
Passed criterion ?
no
yes
Update state of item to S_2
Delete item from system
Update state of item to S_3
Should item be reviewed ?
Transfer item to review pool and update state to S_4
Update state of item to S_5
Update state of item to S_6
GOTO A


ITEM POOL
items 1 - n
WORKING POOL
(5 TO 9 ITEMS)
REVIEW POOL

The Need for Adaptive Control of Aids and Feedback in Problem Solving Instruction

Esther R. Steinberg, A. B. Baskin, & Terri Daling Matthews

University of Illinois at Urbana-Champaign

ABSTRACT

The study investigated an approach to problem solving instruction based on providing aids for improving learners' spontaneous problem solving strategies rather than presenting explicit strategies. The effect of these aids was compared under three conditions: learner, total computer, and minimum computer control of useage. The 78 subjects attempted to solve 8 problems in the initial, randomly assigned condition. On the transfer task they attempted 8 problems under learner control. The aids were helpful, and were used by most subjects. Neither computer nor learner control was most effective. Both conditions of control were facilitative for some subjects and not for others. Factors that contributed to the success and to the failure of each experimental condition are discussed. Results indicate the need for adaptive locus of control not only from one student to another but also for each individual student over the course of learning.

Introduction

In the present study the computer provided subjects with organizational/memory aids to assist problem solving, rather than explicit instruction in problem solving. In contrast to instruction that presents a specific, efficient problem solving strategy, the present approach attempted to maximize learning by beginning with each learner's spontaneous problem solving strategy and providing an aid for improving skills from that point.

If problem solving instruction is to be effective, learners must not only understand the instructor's strategy but also demonstrate learning by using the strategy when no longer required to do so. If the specific strategy is in conflict with that of the learner, he or she may often be unwilling to learn it even though the strategy being taught is more efficient or applicable to a wider set of problems.

A matrix chart was hypothesized to be a useful aid for learning to solve master-mind like problems called Pico-fomi. This problem was chosen because it is typical of a wide range of problems which have a large but finite set of possible solutions, and for which there is no ordered set of solution rules. The matrix chart was chosen as the aid because it can fulfill a useful function for the problem solver without imposing a particular problem solving strategy.

The Game

In the Pico-fomi problem, a 4-digit number consisting of the digits from 1 to 8 is randomly generated by the computer as the target, but not shown to the problem solver. The learner's objective is to figure out the target number within eight steps. The problem solver chooses a 4-digit number and the computer provides feedback that tells the learner (1) how many digits are both correct and in the correct position, and (2) how many digits are correct but in the wrong position. The subject's initial response can only be a guess, but succeeding responses can be built on inferences made from previous responses and related feedback.

The Pico-fomi problem becomes increasingly difficult as the problem solver progresses and generates a multitude of inferences and consequent bits

of discrete information. The problem solver must have access to this information in order to formulate nonredundant hypotheses. To be useful, then, an aid should assist the problem solver in storage and retrieval of all of this information. A matrix chart was generated to provide such aid (Figure 1). The problem solver can record digits that are inferred to be (1) definitely in certain positions, (2) definitely not in those postions, (3) possibly in the target number, and (4) definitely not in the target number. In Figure 1, for example, the feedback indicates that one of the digits is correct but not in the correct position. The problem solver can infer that, regardless of which digit is in the target number, it will not be in the position in which it is located in the present step. The problem solver can mark out those positions on the chart.

The purpose of the present study is to determine how such a matrix chart affects problem solving behavior. Specifically, we are interested in the consequences of learner vs computer control of chart useage. If the learner has control over use of the chart, will failure to use it indicate a preference not to do so, or lack of awareness of its function in problem solving? If use of the chart is under computer control, the computer must keep a record of all possible deductions that can be marked on the chart in order to provide feedback about what information the problem solver has not marked spontaneously. Will the requirement to use the chart, accompanied by this feedback facilitate learning? The hypotheses are that (1) the chart is a useful aid and (2) computer control of chart useage is more facilitative than learner control.

The Experiment

The 78 subjects were paid volunteers who responded to an advertisement in the college newspaper.

The entire study was conducted on PLATO: problem presentation, assignment of subjects to experimental conditions, and data collection.

The three experimental conditions differed in control of chart useage: learner, computer, and minimum computer control. In all conditions the chart was displayed on the screen at all times. In learner control condition subjects were given total control over use of the chart to whatever extent they chose. In the computer control condition, subjects were required to mark on the chart every deduction that could be made. For those deductions that the student did not mark spontaneously, the computer provided the information that the student had omitted, prior to the next response. The minimum computer control condition was a modification of computer control. The computer only required the subject to mark inferences that could be made directly from the feedback on a given step. The subject was not required to mark inferences based on information inferred in previous steps.

All subjects participated in two sessions, one week apart. The initial session consisted of learning the rules of Pico-fomi and the mechanics of using the chart, one practice problem, and then eight problems in the (randomly) assigned condition. The transfer task, presented in the second session, consisted of 8 problems. The chart was not automatically displayed. All subjects had the option of requesting the chart or not, and of using it to whatever extent they chose.

A list of 17 rules of inference, labeled a through q, were programmed into the computer. For every subject, for every step of every problem, the computer recorded which of the rules of inference could be applied, and which deductions the subject spontaneously marked on the chart.

Results and Discussion

Results showed that most of the subjects found the chart useful. In the initial task, all subjects under learner control used it even though not forced to do so. On the transfer task, 59 of the 78 subjects used it to a greater or lesser extent.

Results also showed that computer control of chart useage was not more effective than learner control. Both computer and learner control were facilitative for some learners, but not for others.

Success was defined as solving at least 7 of the 8 problems. There were 26 subjects in each condition. In the initial task, more subjects (21) succeeded under computer than under minimum (15) or learner (8) control (Table 1). In the transfer task, however, only 11 subjects initially under computer control succeeded, compared to 14 under minimum and 14 under learner control. Eleven subjects who succeeded under computer control in the initial task did not succeed on the transfer task. In contrast, all subjects who succeeded under learner control in the initial task also succeeded on the transfer task. Moreover, six subjects who were unsuccessful under learner control in

BEST COPY AVAILABLE

the initial task succeeded in the transfer task.

TABLE 1

Number of Subjects by Changes in Quality of Play

Locus of Control	Chart on Transfer	Performance, initial-transfer task S*-S	S-U	U-S	U-U
Computer	Used	8	7	0	3
	Not Used	2	4	1	1
Minimum	Used	5	3	4	8
	Not Used	4	0	1	1
Learner	Used	6	0	5	10
	Not Used	2	0	1	2

* S=Successful, score 7 or 8
U-Unsuccessful, score ≤ 6

These different patterns of changes in performance under computer and under learner control are best explained by examining the behavior of subjects for whom the assigned locus of control was successful, and those for whom it was not.

For some subjects, computer control was successful: the chart was useful and compatible with their spontaneous problem solving strategies, or else the feedback enabled them to learn rules of inference. For others, computer control interfered in one way or another.

For example, under computer control person RS learned to use the chart to mark simple deductions, and then subsequently to make more complex ones. In the first two problems he did not begin to mark the chart until the third or fourth step (Figure 2). He did not learn all of the rules immediately, not marking rule b for example until the fifth and rules m and n until the seventh problem. On the transfer task he asked for the chart and marked almost every possible deduction.

Some successful problem solvers who were perfectly capable of making deductions wanted some control over marking the chart. Person EK, for example, commented that "having to always stop and write everything on the chart was sometimes annoying when you knew something but didn't realize you knew; you had to write out more than you had actually processed."

For those who succeeded in the initial task and not on the transfer, computer control either minimized the motivation to learn the deductions or facilitated success without understanding. Person DL, for example, waited for the computer to tell him what deductions to make in the first few steps of each problem. The computer provided him 193 messages, compared to an average of 82. Given all of this feedback under computer control, DL was successful on the initial task. He either didn't bother to, or couldn't learn the rules of deduction, and consequently solved just four problems on the transfer task.

Computer control was of little benefit to the four people who were unsuccessful in both tasks. Some of these individuals were able to make the deductions; their deficiency was in generating useful hypotheses, which the feedback did not address. For example, person BB learned to make all possible deductions. Her deficiency was failure to generate hypotheses which could generate a sufficient number of deductions.

Learner control was satisfactory for the 14 subjects who succeeded on the transfer task. However six of them were unsuccessful in the initial task. The latter would, perhaps, have succeeded in the initial task if they had been under computer control for a few problems.

Learner control was of little value to the 12 subjects who did not succeed on either task. Their records showed that some of them marked few deductions on the chart. It is not clear whether they did not know how to make deductions or were unaware of the value of the chart. Others, like person LO, marked the chart adequately but made poor hypotheses so they obtained little new information on successive steps.

Implications for Further Research

The evidence suggests that the matrix chart can be a useful tool in facilitating problem solving. However, the locus of control of such aids, as well as the content of feedback, should be adapted to the learner's needs. Control should adapt not only from one student to another, but for each individual student over the course of learning.

Control of the chart was nonadaptive, always under computer or under learner control. Computer control was facilitative for some subjects and unnecessary for others but compatible with their spontaneous problem solving strategies. Still others found total control abrasive and preferred to exercise control themselves over use of the chart.

Computer control of chart use was not facilitative for all subjects. For some, it seems to have minimized the motivation to learn. If the computer does the thinking, why should the subject learn?

Similarly, learner control of charts was facilitative for some but not

all subjects. The mere availability of the chart is insufficient if the learner does not know why it is useful or how to use it.

The feedback, which was nonadaptive in content and always under computer control, also matched the needs of only some of the learners. Feedback was dynamic in that it was problem-specific. However, it was nonadaptive in that it always provided the same type of information -- every possible deduction that the subject had not marked on the chart. In some instances the feedback was more than the subjects wanted or could understand. Some subjects had a repertoire of rules of inference that was sufficient for success, even though not necessarily efficient. They either did not want to bother with more complex rules, or wanted control over when to learn them and mark them on the chart. For some, the feedback addressed too many rules of inference at one time. These people needed to learn the less complex deductions first before being shown the more complex ones.

The content of the feedback was always the same, what deductions to make. Some learners needed explanation of why the deductions could be made, and different subjects needed different rules explained. Others could make the deductions, but needed feedback about how to make hypotheses.

How might adaptive control of chart useage be applied? First the learner should be allowed to do some problems without the chart in order to familiarize himself with the task. Optional use of the chart should follow. A learner who is succeeding can be allowed the option to use the chart or not, and to ask for feedback as desired. The computer might intervene to present a powerful rule that the learner is not demonstrating. For subjects who are not succeeding, the computer should take control temporarily and gradually shift control back.

This study was supported in part by Army Research Institute Contract Number MDA903-82-C-0277. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

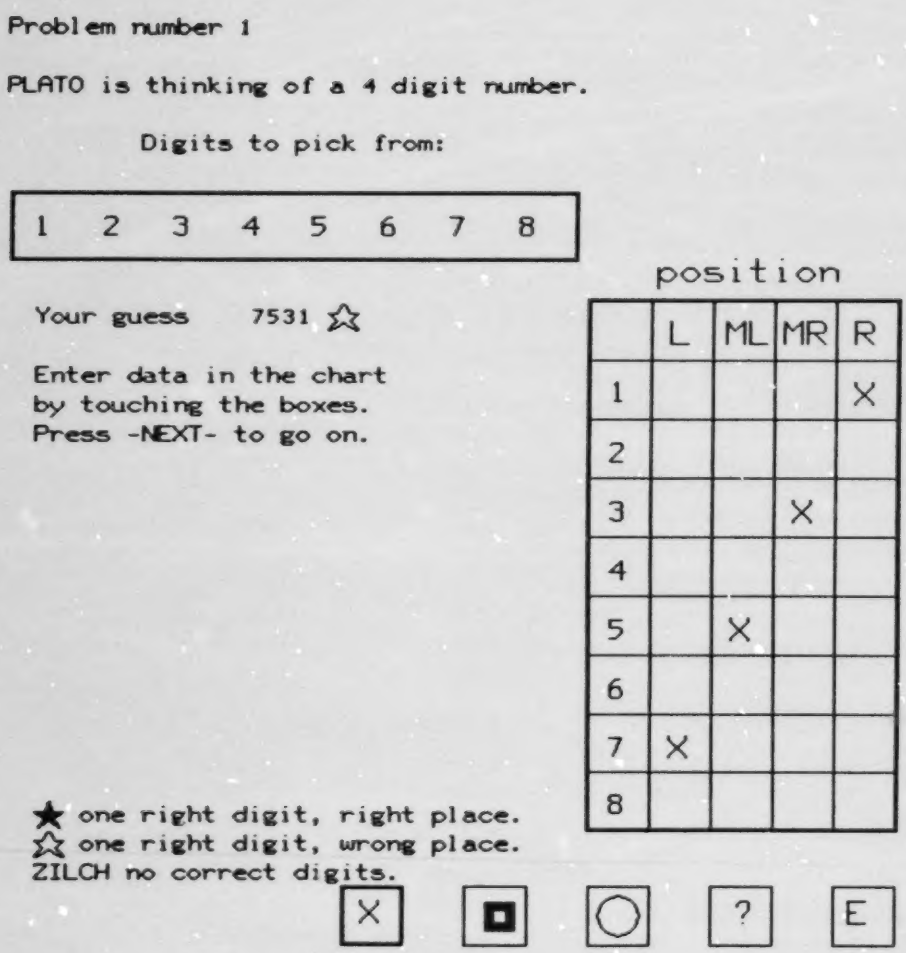


Fig. 1. Example of use of matrix chart to record inferred information in Pico-fomi problem.

Person: RS Condition: Computer control Performance: 8-8

Problem #	Step Number 1	2	3	4	5	6	7
				Initial Task			
				Rules applied by computer (C) and student (S)			
1C	b	o	e	aeh	aceqgi		
1S	-	-	-	e	aceqgi	solved	
2C	a	o	ao	bgqi			
2S	-	-	a	gqi	solved		
3C	a	a	b	-	ajhegqi		
3S	a	a	-	-	ajhegqi	solved	
4C	-	a	a	aemigq			
4S	-	-	a	a i q	solved		
5C	-	b	befgqih				
5S	-	b	bef qi	solved			
6C	-	ae	ae	aemn			
6S	-	a	a	a	solved		
7C	-	b	jgqihmn				
7S	-	b	jgqihmn	solved			
8C	-	b	bknmgqi	bkhgqi			
8S	-	b	k mgqi	bkhgqi	solved		
				Transfer Task			
1C	a	o	aeinm	imnegq	ice	icel	ice
1S	-	-	aei	i gq	ice	icel	ice solved
2C	-	a	ceigq	icelgq			
2S	-	a	ceigq	icelgq	solved		
3C	a	a	a	-	bo	ahgqmnei	
3S	a	a	a	-	-	ahgq n i	solved
4C	b	bkngq	bk				
4S	-	k gq	bk	solved			
5C	b	-	aehn	ahigq	ice		
5S	b	-	a h	ahi q	ice	solved	
6C	a	ao	ae	amnhgq	ae	icegq	
6S	a	a	a	a	ae	icegq	solved
7C	a	-	-	a	a	eihngq	
7S	a	-	-	a	a	i gq	solved
8C	a	a	e	-	al	begqim	
8S	a	a	e	-	al	egqim	solved

Fig. 2. Protocols of rules applied by student RS and rules that could be applied by the computer at each step, in every problem.

BEST COPY AVAILABLE

PROJECT REPORTS AND POSTER SESSIONS

LIBRARY USE INSTRUCTION:

THE CARD CATALOG AND WILSON INDEXES

Rob Aken and Laura Olson

University of Kentucky Libraries

Library instruction traditionally has been presented through a stand-up lecture, sometimes aided by audio-visuals. Students come to us from their freshmen composition classes for one fifty-minute lecture and tour. There is little time to go into the intricacies of the card catalog and periodical indexes, or to give the students hands-on experience with the tools.

We are therefore in the process of writing a computer-based instruction program to show students how to use these tools. To keep from removing the students too far from the actual tools, the program will be used in conjunction with a sample card catalog drawer, a copy of the Library of Congress Subject Headings, and a copy of Social Sciences Index. The students will interact with these tools through explanations and instructions from the computer.

The program is being written in PILOT for a Prime 850 main frame. At present, we have nearly completed the card catalog segment. By the time of the ADCIS conference in May, we will be ready to report on the completed package and its use by c. 300 students.

NAIT'S ALTERNATE TRAINING DELIVERY PROJECT

Mohamed Ally

Northern Alberta Institute of Technology

The two main goals of the project were: (1) to produce self-paced individualized courseware that is transferable to other educational institutions and (2) to reduce the training time for Second Period Apprenticeship students. A variety of instructional technologies were used to help achieve these goals, the main one being the computer.

The computer was used: (1) to manage all aspects of students learning (CML) and (2) to deliver CAI lessons to students. The CAI lessons were produced using "Digital's Authoring Language" on the VAX 11/750. Other technologies used were self-paced print workbooks and audio-visual materials. The completed courseware was delivered at NAIT and another technical institute.

To determine the effectiveness of the project, two groups of students were used. An "alternative group" used the instructional technologies in a self-paced environment and a "traditional group" used the regular classroom instruction.

Results showed that students in the "alternate method" using CML system and CAI lessons along with the print workbooks and audio-visual materials in a self-paced situation performed better than students in the "traditional method" in both completion time and performance.

CENTER FOR COMPUTER LITERACY

Ruth Berggren

John Carroll University

This year, John Carroll University created a Center for Computer Literacy and began a faculty development program for computer literacy with the help of a major three-year grant from the Cleveland Foundation. The program involves semester-long introductory courses for full-time faculty at the university. It is principally, though not exclusively, aimed at departments in the humanities. The program this semester has faculty participants from a variety of disciplines: psychology, sociology, philosophy, history, education, economics, physical education, classics and modern languages, and religious studies.

Besides the differences in application suggested by the diversity of fields among the faculty members, there is a range of other needs. One participant, for example, is legally blind. Two department heads are among the ten participants. A visiting professor from Czechoslovakia (on a Fulbright exchange) is auditing the course.

The faculty have reduced teaching loads for participating in the computer literacy program, so that they will have time to devote to the computer. They receive on loan from the Center a Franklin Ace 1000 microcomputer with a monochrome monitor, a disk drive, and software packages for word processing, data base management, and spread sheet analysis. The blind participant has a Versabraille braille computer on loan from the Cleveland Sight Center and a communications package called "Braille Edit" to link the Versabraille to the Franklin. Hardware and software will be returned at the end of the semester and will be used by the next group of participants in the spring.

The idea behind the Center for Computer Literacy is that, given direction and free time, the faculty will be able to learn to use the computer and apply it to their research and, more importantly, to their teaching. The key is to improve the quality of education at John Carroll by introducing computer applications in classes not traditionally related to the number-crunching stereotype of the computer. Eventually, not just math and science majors, but also liberal arts majors will leave the university having been exposed to the computer and knowing how it can be used in their fields.

INTERACTION OF FEEDBACK AND ANXIETY DURING ON-LINE TESTING

Gerald J. Blumenfeld, Lea Banner, R.O. Blackwood

University of Akron

A series of studies examining the inter-relationships of on-line computer testing, feedback, anxiety, and achievement scores has been underway at The University of Akron. This report focuses on one of those studies that involved elementary school children. Seventy fifth-grade suburban students had been using J. Hirschbuhl's "An Interactive Mathematics Program for Basic Skills and Probem-Solving Competence" as an enrichment program 15 minutes every other day for seven months prior to collection of the data of concern. This program was accessed through ASCII terminals linked to the University's IBM 370.

Sarason's Test Anxiety Scale for Children was administered approximately one month prior to a 20-item mathematics examination. Students were matched according to anxiety scores and randomly assigned to corrective feedback or no-feedback groups. The examination was administered by the computer terminals. Students in the feedback group were informed after every response whether or not they were correct and, if not, which answer was correct. No-feedback students did not receive this information during the test.

The data showed that students in the feedback group obtained an average score of 72.9% correct while those in the no-feedback group obtained an average score of 62.6% correct. This difference is statistically significant. Furthermore, the relationship between achievement scores and anxiety was positive (slope = +.32) for the no-feedback group but negative (slope = -.60) for the feedback group. Essentially, highly anxious students did not do well under corrective feedback conditions.

These findings suggest that the computer's ability to provide immediate on-line feedback is generally helpful to students drawn from this population. However, this assumption must be tempered with the fact that highly anxious students may be hindered somewhat under a feedback condition.

BEST COPY AVAILABLE

THE NATIONAL DEMONSTRATION CENTER

FOR EDUCATIONAL TECHNOLOGY

Victor H. Carr, Jr., Michael Weisberg

Lister Hill National Center for Biomedical Communications

Rapid advances in microcomputer and audiovisual technologies have created new opportunities and special needs for medical educators in planning for the design of medical education programming. Questions about what roles technology will play in supporting medical education, what innovations and improvements are expected and what applications will prove most effective need to be addressed if medical faculty are to be successful in planning for the use of these technologies, and if the potential of these technologies is to be fully realized.

Decisions about using technology should be based on documented criteria, on demonstrations of how technology can support and enhance specific tasks, and on "hands-on" experience with the technologies. Forward looking examples of applications of educational technologies are necessary if planning is not to be obsolete by the time of implementation.

A major project at the Lister Hill National Center for Biomedical Communications, national Library of Medicine, Bethesda, Maryland is the establishment of a National Demonstration Center for Educational Technology. The purpose of the Center will be to address these issues with specific regard to medical education, although transfer to other disciplines is likely.

GOALS

The major goals of a Demonstration Center on Educational Technology will be to provide an applied technology environment in which health professionals have the opportunity to:

(1) Become aware of exemplary technology programs (their attributes and effectiveness) in an impartial setting;
(2) Become motivated to use technology programs appropriately;
(3) Develop an understanding of how to select, use and evaluate technology programs;
(4) Develop the knowledge/skills necessary to select, use and evaluate technology programs; and
(5) Disseminate (share) their acquired experience and information with their peers and colleagues.

This project report will inform ADCIS attendees of developments, progress and status of this center which is currently being planned. By the time of the ADCIS conference in May 1984, a great deal of progress is anticipated including the identification and review of exemplary hardware, courseware, authoring systems and the development of a preliminary demonstration facility.

DEVELOPING AND IMPLEMENTING A COMPUTER ASSISTED INSTRUCTIONAL MODULE FOR DIAGNOSTIC-PRESCRIPTIVE TEACHING

Bert Chiang

University of Wisconsin - Oshkosh

The evolution of the microcomputer into a powerful but inexpensive teaching tool will change the complexion of instructional technology in college classrooms. This project uses microcomputers in the laboratory sessions of the course "Diagnostic Instructional Planning" to improve teaching/learning effectiveness. The primary objective is to develop a computer assisted instruction (CAI) module to teach undergraduate special education majors to analyze, synthesize, and interpret various achievement/diagnostic test scores. The effectiveness of the CAI module is empirically determined by comparing it with the conventional lecture mode of instruction. Parameters that contribute to the teaching effectiveness of the CAI program has been identified by such a comparison.

A series of simulation and case studies are included in the CAI program so that students can actively immerse themselves in environments where they are required to solve complex, real-world problems and where instructors can control the level of the complexity. The major activities of the project involve development, field testing, modification of the CAI module; the implementation of the CAI; and the evaluation and dissemination of the products and processes. The specific topics of the CAI module include simulation of the referral process, error anlysis in the areas of calculations, spelling and behavior observation, and conversion of various types of test scores.

COGNITIVE ABILITIES AND CAI PERFORMANCE AMONG MENTALLY RETARDED ADOLESCENTS

Frances A. Conners

Case Western Reserve University

Investigations into the use of computer-assisted instruction (CAI) for mentally retarded students has been largely limited to determining whether or not certain software is effective in teaching or training these students. Only infrequently have researchers examined the factors underlying such effectiveness.

The present study represents the first step toward a long-range goal of discovering how CAI can be used more efficiently in the education of students with mental retardation and other learning difficulties. Its purpose is to identify the cognitive abilities of mentally retarded learners which are important in learning sight words from a CAI program.

A simple instructional sequence was written for use with an Apple II microcomputer, an Intex Talker speech synthesizer, and game paddles. The instruction was based on principles of paired-associate learning: the visual presentation of a word was repeatedly paired with its auditory presentation.

Twenty students at a developmental center day school underwent pretests of word reading and recognition, four weeks of CAI training, posttests and retention tests, and ten tasks designed to measure cognitive abilities. Twenty additional students, comprising the control group, took pretests, posttests, and retention tests only.

Posttest and retention test scores of the experimental and control groups were compared to determine the instructional effectiveness of the CAI program. Correlational analyses were then performed, using measures of cognitive abilities and of CAI performance as correlates.

COMPUTER LITERACY - A MUST IN TODAY'S SOCIETY

Patricia Veasey D'Souza

The Fogelman College of Business and Economics

The computer became a dominant force in the business world in the late 60's and 70's. Business corporations, government agencies, and other organizations have long depended on the computer to process data and make information for use in decision-making.

However, recent technological developments in the computer industry are rapidly changing and restructuring the society in which we live. Computers now affect the lives of nearly all of us. Whatever you do, wherever you go, you will encounter a computer or computer-delivered services.

As computers become more affordable, they are having a significant impact on education as well as on all other aspects of our lives. Computers are now becoming widely used tools for personal processing of data and information. Today, students in more than 15,000 schools use computers as learning tools. More than 1 million personal computers are installed in homes. The number of personal computers is expected to expand to more than 5 million in the next five years. This means that more than 1 million computers are being sold each year.[1]

The following statement, developed by the Policies Commission for Business and Economic Education, is an expression of philosophy concerning the cooperation demanded in order to educate today's young people:

> All people are affected by the world in which they live. They, in turn, affect the world. Education is the means by which they are equipped to help shape the world in which they live - and live effectively and harmoniously in that world.

Computers are a fact of life. It is important that all members of society learn about computers and understand what they do. Therefore one of the most important teaching functions of computers in education will be the actual teaching about computers. Americans must develop an awareness of the changes that are being brought about by technological advances and the impact these changes will have on their lives as workers, consumers, and family members.

---

[1]Merle Woods, Computer Awareness, The South-Western Publishing Company, 1982.

FEASIBILITY ANALYSIS PROCEDURES AND INSTRUCTIONAL DESIGN PLAN FOR INTERACTIVE VIDEO INSTRUCTIONAL MATERIALS DEVELOPMENT

Allen DeGraw, Hyoja Lee

Florida State University

SUMMARY

The project was sponsored by the Florida Department of Law Enforcement, Division of Criminal Justice Standards and Training. The purpose was to outline a process for analyzing the feasibility of presenting basic law enforcement and correctional officer training by a computer-based interactive videodisc delivery system and to prepare a plan for the design and development of interactive video instructional materials.

The objectives of the study were to define a process which would accomplish the following:

Identify priority tasks for training by interactive video.

Determine cost effectiveness of interactive video training.

Identify factors and strategies which may facilitate the acceptance of interactive video delivery systems.

The project included a model for the subsequent design and development of interactive video training materials that may result from a feasibility study.

EFFECT OF MICROCOMPUTER SIMULATIONS ON COMPUTER AWARENESS AND PERCEPTION OF ENVIRONMENTAL RELATIONSHIPS AMONG COLLEGE STUDENTS

John F. Schar, Rosanne W. Fortner

Ohio State University

Environmental management education emphasizes the interrelatedness of many disciplines in the process of understanding and solving environmental problems. In 1983, 110 college students in an introductory environmental management course participated in a study designed to

1) evaluate the effectiveness of microcomputer simulations in changing specific knowledge and perception of environmental relationship, and

2) increase student awareness of microcomputers and their applications in environmental management education.

The students were randomly divided into two groups. The treatment group used three microcomputer simulations that were selected to support course goals related to presentation of demographic relationships, limits to growth, and ecosystem dynamics. The control group received the same type of information using a traditional workbook format.

A pretest-posttest-control group design was used to administer surveys on Computer Awareness and Environmental Relationship Perception surveys. Random sortment of the test forms resulted in one-half the class taking each survey on the first day, and the randomization process was repeated for a similar testing procedure on the last class day. Neither randomization coincided with that under which students were chosen for treatment or control groups. A Knowledge Subtest of the final exam was administered to all students to assess specific content acquisition in relation to the two information sources.

The Relationship Survey revealed no strong pattern of response shifts on the posttest, and a factor analysis did not produce defensible groupings of items based on the types of responses. Either this method of measurement was not effective, or transfers of learning did not occur as expected.

On the Knowledge Subtest the treatment group outscored the control group at a significant level ($p = 0.02$). Total scores on the final exam and grades for the course were not significantly different between the two groups.

As for attitudes toward the computer, the pretest scores on the Computer Awareness survey closely matched those achieved by MECC researchers in their use of the instrument among eleventh graders. Results indicate that enjoyment declined among the treatment group following the three microcomputer modules ($p = 0.07$). Both groups reported increased efficacy and decreased anxiety related to computer use.

This study has indicated that the short term use of microcomputer simulations may not produce results in terms of stimulating the higher cognitive processes as postulated by others for simulations in general. As a teaching method for factual recall, the computer simulations appeared to be more effective than workbook modules. Interestingly, the computer users indicated feelings that the material from the simulations could have been taught as effectively using standard testbook methods. Apparently this was not the case. Students' perceptions of effectiveness were possibly related to their perception of course goals and to the relative degree of comfort they would have felt in having customary materials from which to study.

As for the attitudes toward computers, enjoyment of computer use may

decrease if use is required and not adequately rewarded. Difficulties encountered by students in interpreting results of simulations or in handling abstract relationships may have contributed to the drop in enjoyment in this study.

EXPERIENCE WITH A SOPHISTICATED COMPUTER BASED AUTHORING SYSTEM

Patrick R. Gardner

Westinghouse Hanford Company

In the November, 1982, issue of ADCIS SIG CBT Newsletter the editor arrives at two conclusions regarding Computer Based Authoring Systems (CBAS): (1) CBAS drastically reduces programming time and the need for expert programmers, and (2) CBAS appears to have minimal impact on initial lesson design. Both of these comments have significant impact on any Cost-Benefit analysis for Computer-Based Training. The first tends to improve cost-effectiveness but only toward the limits imposed by the second.

Westinghouse Hanford Company (WHC) recently purchased a sophisticated CBAS, the WISE/SMART system from Wicat (Orem, UT), for use in the Nuclear Power Industry. This report will detail our experience with this system relative to Items (1) and (2) above; lesson design time will be compared with lesson input time. Also provided will be the WHC experience in the use of subject matter experts (though computer neophytes) for the design and inputting of CBT materials.

WITSCHEM

L. Glasser, J.D. Bradley, G. Brink

University of Witwatersrand - Johannesburg

In 1981, the University of the Witwatersrand introduced a two-year beginning course in Chemistry for non-traditional, educationally-disadvantaged students, who were unlikely to succeed in our traditional first year (freshman) course. Compared to the traditional courses, the new one features a reduced lecture load (3 periods/wk), considerable tutorial activity (2 periods/week, with 8-10 students per highly-qualified graduate tutor), and computer-assisted instruction (1 period/week). Apple II+ and IIe microcomputers are used for the CAI.

Since material adequate in presentation or quantity was not available in late 1980, we developed our own material, setting high standards for ourselves in terms of: correct use of chemical symbolism (upper- and lower-case is not available in the Apple II+ hardware, but is software-generated together with Greek letters, sub- and super-scripts, etc.); protection of the naive user from the operating system; appropriate use of graphics; and careful control of presentation. We now have some 80 programs which are run about 20,000 time per annum on twelve microcomputers which have recently been linked in a network.

This unusual indigenous resource of Chemistry CAI programs has found wide acceptance in some 50 other local secondary and tertiary institutions, but there is a language barrier to its use in the Afrikaans community. With financial support from a local publisher, the material is now being translated into Afrikaans. The experiences of the WITSCHEM project may serve as a guide to other project planners who may wish to become involved in the translation of their material.

MODEL TRAINING PROGRAM FOR RESERVE COMPONENT UNITS

Stephen L. Goldberg

US Army Research Institute - Fort Knox

Reserve Component (RC) units constituting the National Guard and Army Reserve must be prepared to perform their post mobilization missions. The influx of new equipment into the Army has created training problems for RC units to meet this goal. New equipment may not be available or will be in short supply in training locations. Any formal new equipment training soldiers receive will be introductory; there will be no resident experts at RC sites to provide further training. Training on actual equipment may not provide experience performing tasks that are combat related. The Reserve Component training year has a very limited amount of time. State-of-the-art training technology is to be adapted and evaluated to determine its capability to provide soldiers in the RC training environment with training experiences needed to perform their missions using new Army equipment in spite of the limitations discussed above.

A computer-based training program will be developed for use in Reserve Component units for training M1 "Abrams" tank maintenance personnel. The Military Occupational Specialties (MOS) to be covered in the program are: 45K, Tank Turret Repairman; 63H, Track Vehicle Repairman; 45E, M1 Turret Mechanic; 63E, M1 Track Vehicle Mechanic.

Tasks have been slected for training that emphasize troubleshooting and combat maintenance tasks. Computerized courseware developed using the TICCIT authority system will be used to train and simulate task performance for each MOS at skill level 2. Sixty hours of courseware will be developed for each MOS. In addition, supervisor, trainer, and student guides will be developed to support a management system designed to support training in the Reserve Component training environment.

Courseware will stress simulation of trouble-shooting and repair actions based on use of appropriate technical documentation. Videodisc and computer graphics will both be used extensively. The courseware will be evaluated for its technical accuracy, training capability, and transfer performance effectiveness. The transfer studies will be key in determining if soldiers, trained with these materials will be able to perform tasks on actual M1 tanks. After development and evaluation of courseware and management software, the training program will be tried out and its effectiveness determined in appropriate Reserve Component units. The project includes planning for technology transfer and cost-effectiveness analysis.

The project is being jointly funded by the US Army Research Institute and the Training and Doctrine Command. US Army Forces Command has taken an active part in the planning of the project. There is a Memorandum of Understanding between these parties which describes the responsibilities of each in support of the effort.

DEVELOPMENT OF COMPUTER-BASED INSTRUCTION AT THE ARMY ENGINEER SCHOOL

Wayne D. Gray, Christopher P. Werle

U.S. Army Research Institute - Alexandria, VA

DESCRIPTION:

The Army goal for this project is to reduce the length and cost of the Engineer Officer Advanced Course (EOAC) by a series of steps extending through FY90. The Army Research Institute (ARI) has agreed to support this project by conducting a formative evaluation of the development and implementation processes. In addition to the evaluation we have targeted three areas of the development process for active support. First, because of its history of involvement with computer-based training, ARI has developed considerable experience in all aspects of CBT. ARI will provide the Engineer School with advice and assistance in the management and implementation of their CBI efforts. The second involves technology transfer. ARI has agreed to review projects undertaken for other Army sponsors to determine their applicability to the EOAC. Third, the Engineer School has as a high priority that the revised EOAC incorporate state-of-the-art instructional technology. ARI will support this goal by attempting to translate basic and applied research findings (from both Instructional Psychology and Cognitive Science) into applications of use in the EOAC curriculum.

MILESTONES:

FY84 Begin Formative Evaluation
Assist USAES in training and organizing an ongoing CAI authoring cell.
Develop plan for monitoring the quality of CAI courses developed by USAES.
Apply Cognitive Science models of, for example, problem-solving to engineer course design.

PRODUCTS:

(1) Results of Formative Evaluation fed back immediately to the school with recommendations for changes in the development process.
(2) Development of guidelines for formation of CAI authoring cells in an Army environment.
(3) Development of several "model" CAI courses.
(4) Guidelines for applications of Cognitive Science Theories to Army Training.

INTRODUCTION TO THE FUNDAMENTALS OF MUSIC

Harriet I. Hair

University of Georgia

The music programs contained in the Introduction to the Fundamentals of Music were developed under the auspices of the Computer Based Education Project at the University of Georgia. One of the purposes of the CBE Project is to facilitate the faculty development of instructional courseware. These programs were written for delivery by Digital Equipment Corporation's Courseware Authoring System.

These lessons may be used by children (beginning in grade four) or by adults. Initially they will be used in conjunction with the introductory course in music skills and fundamentals which is required of elementary and early childhood education majors at the University of Georgia. They are intended to provide the student with additional tutorial and drill on the materials given in the traditional classroom presentations. These programs could be used with any introductory music textbook.

"The Staff Using the Treble and Bass Clefs." This lesson introduces the staff and the treble and bass clefs. Students learn the definition and function of each of these musical symbols. The lesson consists of tutorial sections, test questions, and drill routines. Students must achieve a certain number of correct responses before proceeding to the next unit in the lesson.

"Identification of the Letter names of the Staff." This lesson is a word game designed to teach the student to identify the letter names of the lines and spaces of the staff using the treble or the bass clef. Contingencies for motivating the student are contained in humorous feedback.

These lessons are being piloted with students of different ages to determine the efficiency and effectiveness of the design. Student scores and attitudes toward the programs will be analyzed and used in the continuing development of these lessons.

REPORTED AVAILABILITY OF MICROCOMPUTERS IN AMERICAN SCHOOLS: YEAR 2

Gary M. Ingersoll, Carl B. Smith

Indiana University

The rate at which society has been moving from an industrial society toward an information society has been accelerated by innovations in microcircuitry and the miniturization of computer hardware. It is not surprising therefore that the spread of computerization should have an impact on schooling. Yet, it is unclear at this point just what that impact is and what trends in computer use in the school are likely. This survey is one of a series of studies of current microcomputer availability and usage patterns in American schools and attitudes that teachers hold toward the advent of high technology in their instruction.

During the Spring of 1982, a national survey of teachers and school administrators was conducted to assess current patterns of availability of microcomputers and other media in American public schools. This survey was repeated during the Spring of 1983 and the results of that second year follow-up survey are the fours of this presentation.

Questionnaires requesting availability patterns, purchase, plans, attitudes and current usage of microcomputers and software were sent to a national sample of teachers and administrators. 68.1 percent of teachers and 66.2 percent of administrators indicated that at least one microcomputer was available in their school. This figure is up from 32.6 percent in the previous study.

Most frequently, those teachers and administrators who report available microcomputers have one, but over three-quarters report having more than two and 16.1 percent of teachers and 13.4 percent of administrators reported that more than 10 microcomputers were in their school, more than double the previous level. On the basis of these figures, it was estimated that in the Spring of 1983 there were approximately 343,000 microcomputers in 57,900 schools nationally, an increase of 185 percent.

Microcomputers are still more likely to be found in secondary schools; 82.1 percent of secondary teachers, 77.3 percent of middle teachers and 59.4 percent of elementary teachers report availability of microcomputers. But, nearly 42 percent of those teachers reporting available microcomputers were elementary teachers. Although the relative availability of microcomputers is lower in the elementary levels, the absolute availability is greatest there.

Among teachers who reported access to a microcomputer, 42 percent indicated they are actual users of the computer. That figure remained stable across all grade levels but is down from the previous year. Most (55.5 percent) teachers report having had some in-service training in the use of microcomputers. But nearly all report a need for some or additional training. When asked to rate their level of familiarity with microcomputers 30.0 percent rate themselves as unfamiliar but wanting help and another 33.7 percent as able to turn it on and load programs. 19.8 percent claim to be able to do simple programming and 6.8 percent advanced programs.

MICROCOMPUTERS AND MATHEMATICS CURRICULUM CONSTRUCTING

Jim Kelly

This project report will briefly cover the second phase of a microcomputer informational system project for K-12th grade mathematics. This phase continues to demonstrate the informational relationship that is beginning to develop between the computer and human beings; this time in the area of curriculum constructing or restructuring.

The presentation will show briefly how computer simulation, computer aided curriculum content algorithm checking, computer bibliographic cross referencing and human understanding can shape or reshape a school mathematics curriculum. Computer aided curriculum designing is economical, fast and will allow for greater individualization of curriculums to meet various socio-economic and linguistic learning environments.

Phase one of the project showed that the 1300 conceptual, operational and notational topics which make up much of the content of K-12th grade mathematics requires only 31% of the available space on a $5\frac{1}{4}$" diskette which an APPLE computer uses. Secondly, by proper coding of the information of K-12th grade curriculums, it has been found on the average that only 23% of the available space on a diskette is needed. The coded information tells which topic appears in which textbook, on what page, for what grade level and how the topic is used. A bibliographic, and scope and sequences program was written to display the information in English, Spanish or French language formats.

HELPING MEDICAL FACULTY AND STUDENTS

OVERCOME COMPUTERPHOBIA

Lloyd A. Lewis

Medical College of Georgia

Three years ago the Dean of the School of Medicine requested the development of a program to introduce faculty and students to computer concepts and to help them understand ways in which computers could benefit them. Goals were not to make computer hackers of faculty and students, but rather to help them become aware of what computers can do for them, how computers effect them daily and how computers will change their lives in the future.

After reflection, it seemed obvious that the key to encouraging students and fculty to mingle with the computer was 'relevance'. The key to relevance, in turn, was applications of computers to medicine. Constraints reared their ugly heads at this time: 1) little equipment and software, 2) little space in the curriculum in which to teach about computers, and 3) busy faculty with little interest in computers. Hence, students would have to be taught in an elective setting; efforts to interest faculty would have to overcome resistance to change and to compete for a busy faculty's time.

Consideration of constraints and the diversity of interests and schedules made it apparent that a multitude of approaches would be the most successful strategy for arousing interest in computers. Thus it was decided that offering electives, workshps, demonstrations, and programming services would provide a sound basis for approaching faculty and students.

Collabortaion with the program in Health Systems and Information Sciences proved to be the easiest way to make a beginning. With Dr. Richard Pogue's help in planning and implementation, a ten hour elective was offered to the freshmen and sophomore medical students. The course began with a brief introduction to concepts of hardware and software but stressed applications for the most part. Medical faculty who were actively using computers in medical education, biomedical research, and patient care were the teachers of the course; hence students saw role models using computers productively. Although the students did not formally evaluate the course, the fact that over 60% % of the equivalent of 25 credit hours of graduate work testifies to student interest.

During the same period, workshops were held for the Dean and his staff at his request. The workshop was a three hour presentation of concepts about hardware and software with the main emphasis on demonstrations of a spreadsheet, a data base management program and a text editing program. Information about computers and programming services were announced and used by a small number of faculty.

Plans for the future include a junior and senior one month elective focusing on the applications of computers in medicine which will be offered next year. These faculty are producing CAI materials or using clinical decision making programs. Students may conduct a literature review of a particular application of computers in medicine. Workshops will be continued for the faculty stressing applications such as test item banking, using an electronic gradebook, using spreadseets, bibliographic retrieval systems, etc. As new medical applications become available, they will be demonstrated, when feasible.

In conclusion, we believe that as

useful applications are demonstrated to the faculty and as we provide useful computer-aided services to the faculty and students, resistance to change will slowly fade. Peer pressure and easing the burden of teaching, research, and patient care will do what words alone cannot.

A MODEL FOR COMPUTER-BASED PROBLEM SOLVING IN APPAREL DESIGN AND PATTERNMAKING

Ruth Marshall

Iowa State University

Many clothing educators believe that fashion merchandising majors need knowledge about garment construction but not hands-on experiences with it. Thus, these educators are looking for alternate delivery systems for this information, systems which would provide students with maximum exposure to garment construction principles with a minimum of skill activities.

The project on which I am presently working seeks to answer the question of whether or not the computer is an acceptable delivery system for garment construction information. The computer simulation developed for this project features the evaluation of construction quality in ready-to-wear blouses and shirts. A series of garments of varying quality are shown on the screen. Students ask for and receive information on construction details, with both visual and verbal responses appearing on request. Students are simulating the role of a buyer in a retail store and are selecting garments for their establishment on the basis of the construction features.

The simulation activity was pretested with a small group of students during Fall Semester 1983; necessary editing and revising was done on the basis of the responses of users to a series of questions about the program. It will be tested with a large group of students during Spring Semester 1984 when the feasibility of using the computer to simulate clothing fit, design, and construction activities will be assessed.

The project, then, has these objectives:

- to develop a computer simulation incorporating apparel design and patternmaking principles
- to determine the feasibility of using the computer for instruction in this subject matter area
- to determine student responses to computer simulations in apparel design and patternmaking courses
- to propose future development efforts based on the results of this study

As the last objective suggests, future development efforts will be determined from the results of this study.

COMPUTER TESTING IN AN INTERNAL MEDICINE CLERKSHIP

MARY MEDAL, LYNN KERBESHIAN, REED T. KELLER

UNIVERSITY OF NORTH DAKOTA, SCHOOL OF MEDICINE

In 1979, legislative action expanded the two year North Dakota curriculum to include a final year for electives at community hospitals within the state. The 1981 Legislature additionally authorized the teaching of the third year in North Dakota with 30 students in Fargo and 10 in Bismarck for the 1983-84 program.

Included in the third year is a twelve-week required clerkship in Internal Medicine. As students are located in two cities, the experience was structured around a core curriculum utilizing self-study materials and computer testing. Briefly, each division (Cardiology, Clinical Pharmacology, Dermatology, Endocrinology, Geriatrics, Gastroenterology, Hematology) developed self-study packets which included objectives and references. Students were to study the packets in any order and take a test on the contents when ready. Divisions submitted corresponding test questions based upon the learning materials. Acceptable formats included one-best-answer, multiple choice, true-false, or Type K. Questions could be keyed into the program in two ways - by objective or as a divisional pool. Tests entered by objective would be generated with a preset number of randomly selected questions per objective up to 25 items. When entered as a pool, any 25 items were randomly selected.

Procedurally, students signed on to the system with a unique identification number and selected a divisional test. The program allowed students to take as many tests as desired at one sitting but locked them out of a particular division for 24 hours if 70% accuracy was not achieved. The correct answer was displayed following each item. After completing a test, students received a percentage score and a pass/fail message. Upon failing a test for the second time, students were referred to the respective Division Chief for consultation and oral testing. Scores and item analysis information, e.g., which options were selected, were permanently stored for later retrieval.

Equipment installed included three Zenith Z-29 terminals in Fargo and two in Bismarck. The library staff oversees the operation and terminals are available when the libraries are open. User support is facilitated by use of a hot-line number to Grand Forks with evening and weekend coverage provided by work-study students.

The Internal Medicine Testing Program has been used to determine whether educational objectives have been mastered and to ensure that selected standards are being maintained. Although the primary emphasis during the clerkship is on clinical activities, the tests serve to motivate students to review and acquire information necessary for the proper care of their patients. The system has greatly enhanced efficiency in terms of test preparation, administration, and record-keeping.

Problems have mainly centered around "computer malfunctioning," especially in Bismarck where a communications link had to be established with NDSU in Fargo through Central Data Processing. Steps taken to enhance the operation include posting notification that students should report any difficulties immediately (down time in Bismarck dropped from 17 percent to 9 percent when hours of unreported problems were discounted), staff orientations, and adjusting operational parameters. In addition to continuing item revisions, future developments include the addition of feedback for each option.

MANAGING AN INTERNATIONAL PROGRAM

BY COMPUTER COMMUNICATIONS

Karen Miller

University of Southern California - Los Angeles

The idea of using computers and an international satellite-telephone line connection to manage in an instructional system came as the result of a need of the School of Business at the University of Southern California to improve instructional communication with the Madrid, Spain, Center for International Studies. Many USC students attend this center to gain international exposure to business in another country and need to take courses in their major field in order to complete their undergraduate degree.

The challenge of setting up the system in another country involved making arrangements with the Spanish government, securing appropriate hardware, and selecting an international communications network. Telenet was the network we selected. It has proven to be a very inexpensive way of delivering, storing and retrieving a high volume of messages and instructional materials.

We dial up our on-campus mainframe and access our file, which has many user names and passwords to allow for confidentiality. The Madrid Center uses a local telephone number which gets them into Telenet and carries by satellite and dedicated phone lines data packets. These packets end up in the same mainframe message files. We have had great success with both a 300 BAUD rate and a 1200 BAUD rate. We had to purchase European made modems because the Bell standard is not used in Europe. They use a tone system called CCITT, and of course the voltage is different from ours, too. For those two reasons, plus heavy import duties, it was simpler to buy European equipment made by American manufacturers. At USC we already have the Telenet software in our mainframe, and we have terminals in most offices on campus, so the system is very easy to use.

The advantages to our program are many. One is money. We are using the computer link instead of the mails and the telephones. This has reduced our cost of communicating from hundreds of dollars per month to an average $50-$100 per month. From Europe the connect time for Telenet is about $7 per hour and $1.55 per 1,000 packets. Another advantage is that the users of this system could learn it quickly and really enjoy it. They feel lost if the computer crashes, almost as though they lost a friend. But the main advantage is that the students are learning more as measured by comparative achievement tests.

We are presently working on a microcomputer file transfer system through Telenet that will allow us to dump entire diskettes across the more than 10,000 miles instantly. That way, we can word process our papers before they are sent. We will be happy to share the information on how this is done.

COMPUTER TECHNOLOGY PROGRAM

Jim Pollard

Northwest Regional Educational

The Computer Technology Program has been involved in developing methods and materials for the use of computers in education for more than a decade. The program has several current projects which have a common theme of assisting educational agencies to make the most effective use of technology, both hardware and software. The Laboratory continues as a national leader in the evaluation of microcomputer software and the dissemination of courseware information. New projects focus on communication technologies and the innovative use of these technologies in schools.

The MicroSIFT project operates as a clearinghouse for information about microcomputer-based educational applications. The project includes a network of 27 educational institutions and consortia (SIFTnet) which collects and summarizes reviews from experienced computer users. The institutions represent a variety of geographic, ethnic, minority and socio-economic student groups. The Program coordinates the reviews at these institutions and distributes the information through a network of dissemination sites.

The Program is also involved in an automated data base which provides state-of-the-art information about the instructional applications of computers in elementary and secondary schools. The system, Resource in Computer Education (RICE) has been in place for over a year and contains courseware descriptions, courseware evaluations, and publisher data. The data base may be searched by such parameters as grade level, subject, type of computer, and publisher.

The Program's Technology Center provides training throughout the nation on such topics as Business Applications of Microcomputers, LOGO, and Special Education and Computers. Through products like the Computer Literacy Guide for Administrators, the Center makes its curriculum available to all educators.

The Computer Technology Program has begun a focus on communication technology in education. The Program is demonstrating the effective use of these technologies through regional networks, electronic dissemination of products, and the electronic gathering of data for state and local agencies. The Program is taking an active role in the development of the potential of Cable Television for interactive long distance computer, visual and audio communication.

MATHEMATICS ENRICHMENT PROGRAM

Thomas Ralley

Ohio State University

For the past four years the Mathematics Department at Ohio State has conducted an enrichment program, New Horizons, for mathematically talented junior high school students. During this same period, there has been nationwide discussion at the university level about changes in the undergraduate mathematics curriculum that are needed to prepare students for a world in which computers play an increasingly important role. The presentation will report on the discrete mathematics and the associated microcomputer activities done with New Horizons participants in an effort to identify topics that might be included in a new curriculum and that provide opportunities for stimulating problem solving activities. The mathematics involves combinatorial graph theory and algorithms for solving a variety of graph theoretical problems. Microcomputer activities involve methods for displaying graphs and for illustrating and studying the algorithms. The presentation should be of interest to educators searching for interesting topics in discrete mathematics that make effective use of microcomputers.

DEVELOPMENT OF INTELLIGENT VIDEODISC

TRAINING MATERIALS FOR SCIENCE TEACHERS

Thomas C. Reeves, James Okey, James M. King

University of Georgia

OBJECTIVE

The purpose of this project is to design, develop, implement, and evaluate a stand-alone, exportable interactive microcomputer/videodisc program for training precollege science teachers to integrate computing into their instruction.

BACKGROUND

Rapid change and widespread use of computers in business, industry, government, and higher education place heavy pressure on schools to graduate students with the skills to play productive roles in an increasingly technological world, but many teachers lack the knowledge, skills, and motivation to integrate the products and prcedures of technology into their classrooms. Many colleges now include some computing in their teacher training programs, but this provides only a trickle of new talent into a field that needs to rapidly incorporate computers into both the content and methods of schooling. Inservice school teachers need the means to rapidly, effectively, and economically upgrade their abilities to use computers in instruction. This need is particularly acute in science education in which a shortage of qualified teachers is worsening at the same time that computer applications in the sciences are increasing.

METHODS

This project makes use of intelligent videodisc (videodisc controlled by a microcomputer) materials to deliver instruction to teachers in on-the-job settings. Independent, interactive training materials, using the best technology available, are utilized so that the teachers learn new computing skills on the same technology they later use to deliver and manage instruction for their students. Intelligent videodiscs are used in four roles in this program, tutor, tool, tutee, and tour guide. The program is a tutor by providing instruction on a full range of teacher responsibilities including testing, record keeping, reporting, preparing materials, and delivering instruction. As a tool, the intelligent videodisc program provides science teachers with a wide range of resources for instruction and classroom management including word processing, problem solving, data processing, and simulated laboratory experiments. As a tutee, the program provides teachers with opportunities to "tutor" or program the computer to accomplish specific instructional objectives. Finally, as a tour guide, the program provides science teachers with "microworld" simulations in which there are numerous opportunities to solve instructional and management problems applicable in their own classrooms.

STATUS

In December 1983, a prototype videotape for college level instruction will be shipped to the 3-M Corporation for videodisc mastering. Pilot testing will be conducted at the University of Georgia beginning in January 1984.

BOSTON COMPACT CAI/CMI INITIATIVE

Mary Russo

Boston Public Schools

The Boston Compact CAI/CMI Initiative is a partnership among Boston businesses, the school system, and city government to support computer-based instruction in basic skills for 400 twelfth grade students in five of Boston's seventeen high schools to develop competencies in math, reading, writing and problem-solving required to get, keep, and advance in jobs in the Boston area. Under its terms, the business community agreed to hire high school graduates in permanent jobs provided the schools prepared them with requisite skills in reading, math, and problem-solving. Using VT100 minicomputers donated by Digital Equipment Corporation, Apple microcomputers, and integrated learning program and word processing software, students augment basic skills development with computer awareness and simple computer use skills needed in the increasingly automated workplace. Students were selected for the project based on scores on the Metropolitan Achievement Test: Mathematics, the Degrees of Reading Power Test, and citywide curriculum-referenced tests. The content of thirty CAI programs for reading, mathematics and language arts have been aligned with 104 reading, mathematics and writing competencies from the school system's citywide reading/language arts and mathematics curriculum objectives and form the core instructional goals of the project. The course of instruction, individualized through the use of student contracts, features continuous assessment, 75 percent of which is computerized, as the main planning tool. A demonstrate- practice-apply learning model is the framework for instruction.

Major goals of the project include gains in math and reading for 85 percent of the students in the project to meet minimum competency standards established by the school committee and placement in jobs requiring computer use of skills in hospital and health care settings, retailing outlets, automated offices, and computerized automobile inspection. Project materials include a project manual, a management manual, and a training packet for staff development.

APPLICATION OF AESTHETIC LEARNING MODELS FOR THE APPRECIATION OF COMPUTER ART

Penelope Semrau

Shaffer & Shaffer Applied Research and Development, Inc.

This project is concerned with identifying aesthetic concepts for developing software related to the appreciation and creation of computer art. This research will provide a foundation for the eventual development of an aesthetic learning model for use with creative graphic imagery.

In 1965, the first exhibit of computer-generated visual art was organized at the Studio Gallery of the University of Stuttgart (Germany). Increasingly computers are being used in education for the teaching of graphic art concepts and as a medium for creating art. There are several issues related to art education that are of major concern to this report: (1) is there a new and/or different aesthetic criteria which needs to be emphasized in computer imagery, and (2) how can we educate people about the aesthetic qualities of computer images? (Madeja, 1983).

Aesthetic education has been defined as "instruction in ways of appreciating works of art" (Smith & Smith, 1978). Art appreciation extends from knowledge about art to feelings about art. Aesthetic education models diagram or describe stages, levels, or a series of sequences that the learner may proceed through in responding to works of art. Various aesthetic education models can be examined to identify categories that are appropriate for the development of art appreciation. For example, Broudy's model (1966) was divided into aesthetic perception and aesthetic criticism. Aesthetic perception included observing the "sensory properties" of shape, color, texture, etc.; identifying the "formal properties" of theme, balance, unity, etc.; recognizing the "expressive properties" of mood, ideas, etc.; and finally, noting the "technical properties" of how it was created - the brush stroke or pen line. In the area of aesthetic criticism, Broudy included recognizing the "historical perspective" of style, period, and culture; interpreting or "recreating" (stage of recreation) the artist's intent vicariously; and, judging or estimating (judicial stage) the value of the art in relation to a criteria of formal excellence, significance, and truth. Broudy's model has implications for aesthetic education in computer imagery.

In researching the literature in art, issues and problems have been identified within aesthetic education. Critics have noted the elitist attitudes frequent in the aesthetic education literature. Greater emphasis needs to be placed on aesthetic inquiry as well as historical and critical factors (Lanier, 1983). In addition, computer art has not been considered within the aesthetic education framework.

An examination of the literature in educational computer technology has revealed a limited concern with visual art education. Nygard and Ranganthan (1983) suggested that opportunities be provided in which the student works with the data represented in visual displays. However, "visual display" was defined or limited to illustrations such as graphs or charts, and did not include the more complex, imaginative phenomena involved with works of art.

As educators become involved with the technology of computers and the aesthetics of computer art, the issues of (1) what factors need to be identified to develop an aesthetic literacy in

this media, and (2) how will this aesthetic literacy be developed - needs to be addressed.

COMPUTER-ASSISTED INSTRUCTION IN LABORATORY SCIENCE

THE GEORGE WASHINGTON UNIVERSITY MEDICAL CENTER

Sylvia Silver

The George Washington University Medical Center

This project, funded by the W.K. Kellogg Foundation, is aimed at developing computer-assisted instruction for the continuing education and retraining of laboratory science practitioners. Development of computer modules in four major areas of clinical laboratory science has begun. The work involves textual content specialists, instructional design personnel and computer programmers. The modules contain text, color graphics and animation augmented with user interaction. The modules are:

1. Therapeutic Drug Monitoring
2. Histocompatability Antigens
3. Platelet Function Studies
4. Determination of Coagulase
5. Basic Renal Function
6. Cytodiagnostic Urinalysis

These modules have been developed on a dedicated microcomputer using an authoring language. Future plans include the addition of material for videotape and videodisc sources in an interactive mode.

THE CERL CLUSTER SYSTEM

Bruce Arne Sherwood

University of Illinois - Urbana-Champaign

The development of the CERL Cluster system is an attempt to provide the best features of both mainframe and micro-based computer-based educational systems, at lower cost than previously attainable, and with the aim of being able to exploit diverse hardware through machine-independent system software.

The system is a local area network designed specifically for computer-based education, and having the unusual property of permitting large, complex lessons written in μtutor to run in small station memories. Lesson segments are read into station memory on demand, as needed, through a 2.25 megabit/sec packet network. A hub computer (presently a 68000-based Motorola system) reads lessons from hard disk into its main memory and delivers lesson segments upon request through the network. Several different micros have been used as stations, including the CDC IST and Viking, and the NEC 8800 and 9800. It is intended to exploit other micros as well, and DuoSoft Corporation has already run the same ascii-based μtutor in stand-alone mode on the IBM PC and Toshiba T300.

The Cluster system provides full authoring capabilities in μtutor, including graphics and character-set editors. There is a new, simplified sign-on procedure and a sophisticated router for guiding student study and reporting to instructors.

EXPERIMENTAL PSYCHOLOGY PEDAGOGY: OBESITY SIMULATION

John Slaughter and Thomas Rywick

State University College - Fredonia, N.Y.

An experimental psychology simulation study was designed using the research data in the area of obesity. The simulation scenario involved a subject in a waiting room prior to an experiment. In the room there are bowls of peanuts, chocolate kisses and finger sandwiches. The dependent variable is how many units of food the subject consumes during a 45 minute waiting period.

The research methods instructor can send students to the literature to read past research and generate testable hypotheses regarding variables affecting obesity. Within the simulation the student can choose to manipulate any of nine potential independent variables. The variables selected can be manipulated over a set range and those not manipulated will assume a default value.

The presentation will discuss the model itself and the use of experimental simulation studies in teaching.

NCR PILOT

Timothy W. Spannaus, Pat O'Brien

Creative Universal, Inc.

NCR Pilot is an enhancement and modification of Western Washington University Pilot, designed to be transported to a wide variety of NCR operating environments.

In the initial implementation, Pilot was defined as a tool in the Software Development Environment (SDE). As such, it was immediately available in three major operating systems. Later developments provide source-code compatibility from the DM-V personal computer to the largest business mainframe.

In providing broad compatibility, NCR has had to deal with difference in hardware capabilities, disc file operations, and so on.

The result of the effort will be the ability to run a wide variety of courseware in essentially the same form on virtually any NCR machine.

COMPUTER-ASSISTED, VIDEO-TAPE LEARNING MODULES

Louise Timmer-Hawck

California State University - Sacramento

This project demonstrates a new advancement in self-paced, individualized, learning modules. It brings a new dimension to the present use of CIA (computer-assisted instruction). For the first time, video tape instruction has been developed to be used with computer-assisted instruction. The system is easy to use for the student. It allows video-tape instruction to interact with a computer giving the student instant questioning as he/she progresses through the programmed instruction module. The primary focus is on mastery learning, permitting repetition of content until mastery has been attained. The video tape is synchronized with the computer to give instant replay of content missed in the question and answer portion of the computer-assisted instruction. The advantages of the video-tape, computer-assisted instruction include greater dramatization of content, synthesis and application of concepts, theories, facts and events, intermittent questioning with instant replay of information not well understood, and live-action instruction focusing on strengthening the student's ability to make correct decisions.

The authoring system is quite simple. The language is teacher-language which brings the instructor step by step through the programming of a learning module. The video-tape can be designed by the instructor or adapted from a commercial video-tape. The advantages of this authoring system include flexibility and adaptability to the classroom needs. The instructor can design a learning module for content that is not well developed or addressed in a text, content that requires mastery, learning that must be completed prior to class, visual and graphic representation of a concept not well explained in words, and visual development of a concept, idea, or theory involving movement or progression. The teacher can program answers not only for multiple choice, true-false, but also short answers, and essays. The computer printout can be used along with this system to correct tests, compute the number of errors made by the student, and store the tests for later use in profiling the student's learning difficulties throughout the semester.

The equipment involved with this system is very basic. An Apple computer, a video-tape recorder, and an interface monitor that allows the two systems to interact. The discs and video-tape are economically priced and cost only pennies when several students use the learning modules.

Research is now being conducted at the California State University, Sacramento, Division of Nursing, comparing three traditional learning modules with the video-tape computer-assisted learning modules. There are three computer-assisted, video-tape learning modules being investigated. They are The Endocrine System, Infant Health Assessment, and Techniques of Bathing the Newly-Born Infant.

EXTENSION HOME ECONOMISTS LEARN HOW TO DESIGN CAI BY CAI

Emily S. Wiggins

Clemson University

It seemed a "natural" to teach Designing CAI using CAI. That is why the Extension Family Life Specialist, Emily S. Wiggins, at Clemson University in Clemson, South Carolina, taught three groups of Extension Home Economists how to design Computer Assisted Instruction using three different instructional methods. One group learned principles of designing CAI through a traditional workshop, another group learned through some lecture and some CAI.

One Home Economics "mini" lesson appropriate for CAI is being designed by each participant. Each "mini" lesson will be evaluated by a subject matter specialist as well as by a panel of CAI experts. A validated evaluation instrument will determine if CAI can be used to teach the design principles of CAI as effectively as can a traditional lecture. Costs of the three methods of instruction will also be compared.

SURGICAL SPECIALTY CURRICULUM DEVELOPMENT AT THE MOUNT SINAI SCHOOL OF MEDICINE

David P. Yens, Joan T. Peters

Mount Sinai School of Medicine

The purpose of this project is to develop self-instructional, mastery-based, computer-evaluated curriculum materials for the surgical specialties of Anesthesiology, Neurosurgery, Orthopedics, Otolaryngology, and Urology. These materials primarily address the cognitive knowledge and skills from each specialty that are essential for the general practice of good medicine. Study of the self-instructional materials is done concurrently with or immediately preceding the clinical week spent on the particular specialty. Study materials and the testing system are located in the Levy Library of the Mount Sinai School of Medicine. The study materials for each specialty consist of not more than four hours of audiovisual content (slide-tapes, videotape) and may also include readings from assigned texts or articles.

The testing activities utilize a computer-assisted testing system on the IBM Personal Computer. When the student completes study of the assigned materials, he/she signs on to the PC, identifies him/herself, and selects the test to take. Items for the 30-question test are randomly selected from a large pool of items specifically prepared for this program. Immediate results (percent correct and whether the test was passed or not) are furnished upon completion of the test. If the test is failed, the student can retake it after added study. A different set of questions is generated for each test, so every student sees a different test and repeated tests consist of different questions. Complete records of test performance are stored to permit comprehensive analyses of each item on which revisions can be based. Thus, the quality of the item pool will be improved over time.

Initial responses by students to the self-instruction and self-testing system have been good.

Future plans include: 1) Improvement of the item pool by using more problem-solving type items; and 2) Development and use of computer-based simulation materials in the surgical specialties.

INTERACTIVE COMPUTER GRAPHICS AND THE TRANSITION FROM CONCRETE MANIPULATION OF OBJECTS TO ABSTRACT THINKING IN ELEMENTARY SCHOOL MATHEMATICS

DONNA BERLIN, ARTHUR L. WHITE

OHIO STATE UNIVERSITY

This project was designed to explore the learning model which suggests that a concept is acquired first through manipulation of concrete objects followed by transformation of the concrete objects into semi-concrete representations and followed by internalization of the concept through abstract representations (i.e. symbols, letters, numbers). Microcomputer simulations of mathematics laboratory activities have been developed and field-tested for the purpose of exploring the transition role of the computer simulations.

The study involved elementary school children consisting of a predominately black population and a predominately white population. The children in these schools used the computer simulations over a 6 week period. The simulations include graphic representations of the following concrete activities: geoboard shapes, geoboard spatial orientations, pegboard and colored cube patterns, pegboard and colored cube symmetry. Comments and suggestions from teachers and students were collected and summarized for the purpose of revising the computer activities. A measure of student understanding of visual discrimination, spatial orientation, and pattern recognition was developed and piloted during the computer activity field testing.

At this point in time field test data has been collected from Lincoln Heights Elementary School in Cincinnati, Ohio and the Tri-Valley School District in Dresden, Ohio.

AN ELECTRONIC LOGBOOK/EVALUATION SYSTEM

FOR CLINICAL TRAINING

L. BRENNER, J. COWAN, T. WACASER

UNIVERSITY OF ILLINOIS, URBANA

Accountability for the content and quality of undergraduate and postgraduate medical education is increasingly emphasized by certification agencies. The nature and size of a trainee's patient load and the opportunity to perform procedures significantly reflect the content and to some extent the quality of training. Many trainees are currently required to collect this documentation through a hand written log of their activities. They record the number of patients seen, the problems or diagnoses associated with each patient and the procedures performed. Compliance is traditionally poor and hand collation of logbook information is laborious. Trainees see the process as time consuming and having no impact on their subsequent clerkship experience or evaluation. Nevertheless, this information is increasingly required by certification authorities and by hospitals evaluating medical staff candidates.

We have developed an Electronic Logbook to track clinical experience and are implementing the system in January of 1984 with internal medicine clerkship students at the University of Illinois, Urbana. The Logbook is programmed in Tutor and runs on the University of Illinois PLATO system. PLATO terminals located at each hospital and at the University of Illinois will be used by trainees and their instructors. After the field test, the system will be implemented for clerkship tracking at Rush.

The Electronic Logbook provides an ongoing record of each trainee's experience which may be reviewed by the course director throughout the training period. This timely information may be used by the instructor along with electronic bulletin board communications to plan supplementary activities filling experiential gaps in the training program. The electronic logbook also includes final course evaluations. Most importantly, the Electronic Logbook also impacts trainee evaluation through a link to an item banking and test delivery system. Periodic mastery tests are automatically tailored to the recorded experience of each trainee. By reducing the overhead in data collection and using this data to directly impact both training and evaluation; student compliance, the educational program, and accountability are expected to improve.

The development of the Electronic Logbook and its link to the MTA Evaluation System presents a unique opportunity to both objectively test the varied experiences that characterize clinical training as well as to establish a commonly agreed upon core of concepts and activities. The definitions that characterize an Electronic Logbook project tie training programs at different sites together. Faculty can organize, review and revise the core objectives on an ongoing basis; and they will do so because agreeing on a common core will not detract from the importance of opportunistic experiences that will continue to differ from site to site. Students will be documenting varied experiences and will receive credit for mastery of related topics just as with the core material.

Clinical testing has previously been associated with common experiences, detracting from the richness and variety of clinical training. The tracking and evaluation system will together solve many institutional and

professional dilemmas that currently plague training programs and emphsize the importance of the investigative skills of research and analysis.

APPLICATION OF HAND-HELD COMPUTERS IN TRAINING

LARRY D. FRANCIS, GERARD CHEVALAZ, JOHN MERRILL

BATTELLE - COLUMBUS LABORATORIES

ABSTRACT

In the past two years researchers at Battelle have conducted three studies on the feasibility, application, and implementation of hand-held computers in training. The first study undertook to examine the feasibility of using hand-held computers for training purposes and to review the available hand-held computers on the market. Results of this study demonstrated that a significant number of training situations could be supported and enhanced by available hand-held computer technology. Several hand-held computers were identified as offering particular promise in this area.

The next two studies evaluated the success of two applications of the hand-held computer technology to training. The style of the training in the studies differed greatly. In the second study the courseware consisted of conventional drill and practice number conversion on the Hewlett-Packard HP-41CV. The third study developed and analyzed a training package of simulated troubleshooting tasks on the Quasar HHC. The problems, successes, and usefulness of these unique training devices were pinpointed. All three studies were sponsored by the U.S. Army Training Developments Institute.

CHEMICAL FORMULA ENTRY AND DISPLAY IN WITSCHEM

L. GLASSER, J.D. BRADLEY, G. BRINK, S. LANGLEY

UNIVERSITY OF WITWATERSRAND, JOHANNESBURG

Keyboard entry of linear chemical formulae (e.g., Hg_2^{2+}) is a difficult task on a typewriter, but more difficult on a microcomputer because of the lack of special characters, such as sub- and superscripts, or script (as in C). This problem is compounded with the Apple II+ microcomputer which has no proper shift-key, and with naive users who cannot be expected to learn or remember special key sequences.

The problem has been largely solved by using software generation of characters (with Apple's HRCG), specially created character sets (the standard ASCII set together with our CHEM set), and a limited range of chemical formulae, which permits application of some rudimentary "artificial intelligence". It should be noted that the problem is more difficult than that of text-processing (for printing) since display decisions must be implemented immediately each character is entered.

Rules have been established which permit correct interpretation (as upper-or lower-case) of alphabetic characters which have been entered from the keyboard, according to their relation to previously entered characters. Numeric entries are always displayed as subscripts. Superscripts only appear in our system together with charges, e.g., A^{3+}. These are entered by repetition of the sign, viz., AL+++; the coefficient of the sign in the screen display is updated as each sign-character is entered at the keyboard.

Some problems remain in this simple scheme; it is not possible to distinguish between, for example, CO and Co and similar combinations, or some unusual combinations such as NaL. . . and NA"1". . . . These are either treated as special cases in the context of the problem, or avoided.

It is believed that a completely comprehensive "artificially intelligent" scheme without a shift-key is not possible. However, all new microcomputers of any consequence now have an operative shift-key so that this difficulty largely falls away.

Display of two-dimensional projection formulae, e.g., is more complex, and is only initiated from within a program. Data items are used which specify the shape number, and - and -coordinates of its origin. The data items are generated by our utility FORMDRAW which simultaneously generates the data and displays the shapes on a grid. The formula may be edited by re-tracing steps backwards through the sequence of entries.

INTRODUCING PICTURE CLAUSES:

A TUTORIAL PROGRAM FOR COBOL STUDENTS

KAREN H. GUTHRIE

UNIVERSITY OF TOLEDO

This poster session introduces the first in a series of tutorials developed for an undergraduate course in COBOL at the University of Toledo. The purpose of this first tutorial program is to teach students about the use of PICTURE clauses. PICTURE clauses describe characteristics of specific storage areas in a computer's memory. These areas may store input/output information or they may serve as temporary locations for processing numeric data. The storage areas vary both in size and in the type of information stored. Therefore, the PICTURE clauses describing these areas must also vary. The contents of a PICTURE clause depend both on the type of data in storage and on the way in which this data will be processed.

In the tutorial program, students first learn about non-edited PICTURE clauses. These PICTURE clauses are typically used to describe input or processing areas. Students learn how to determine the length or "size" of a storage area and the meaning of symbols representing alphanumeric, numeric, and alphabetic data as well as symbols representing implied decimal and signed fields. Next, students learn about edited PICTURE clauses. These PICTURE clauses describe data for printed output. Fixed insertion, floating insertion, and replacement characters are explained. Students learn how to determine the length of edited fields. This program also describes the relationship between the MOVE statement and the type of data described in the PICTURE clause.

The program consists of a series of short paragraphs. Each paragraph is not more than four sentences long and is followed by a multiple-choice or "fill-in-the blank" type question. Students type in an answer and are notified immediately whether or not they have answered correctly. The questions and answers sessions help students learn to differentiate between valid and invalid PICTURE clauses as well as between valid and invalid MOVE's. The software is written in BASIC and has been adapted for use on Apple II+, PDP 11, and IBM 4341 systems. The program should be of interest to secondary, junior college, and university educators.

TEACHER IN-BASKET: STUART AND JUANITA

S. LANGHAM - JOHNSON

BOWLING GREEN STATE UNIVERSITY

This computer assisted instructional program was developed for use with undergraduates in teacher education, within the teacher education redesign plan. The redesign plan mandates increased time spent in field work and clinical activities that simulate life in the classroom.

The Stuart and Juanita programs provide simulation of teacher interactions with parents, dealing with students who have special problems. The Stuart program involves the student teacher in deciding how a teacher should respond to a suspected case of child abuse. (Wrong responses include getting punched in the face by an irate parent, if the student decided to accuse the parent of child abuse.) The Juanita program introduces ideas about possible interpretations of standardized text scores, comparing scores on aptitude tests and classroom grades, for an underachieving migrant worker's child. This program demonstrates the potential of computer assisted instruction for simulating teachers' problems in a clinical setting.

Apple II with disc player and display.

ECOLOGICAL CAI 112

S. LANGHAM - JOHNSON

BOWLING GREEN STATE UNIVERSITY

Ecological CAI 112 is a computer assisted instructional program that includes text presentations, multiple choice items, review, practice branch programs and integrated video segments. Designed for undergraduates in EDFI 302, Introduction to Educational Psychology, the program is based on theories of behaviorist and ecological psychology (Jacob Kounin, 1969).

In this program, students look at video segments drawn from the Child Variance Series. These include naturalistic classroom settings and common problem-behaviors encountered in classrooms. Students read text about the psychological theories and interact with the program, choosing some course of action in relation to student misbehavior occurring on the videotape. The ecological perspective engages the student in identifying environmental and situational contexts within which student learns to respond to students' problem behaviors in the classroom without disrupting the on-going flow of instruction.

Apple II with disc player, video screen and display.

TRAINING TEACHERS TO USE MICROCOMPUTERS

HELEN LEE

INDIANA UNIVERSITY/PURDUE UNIVERSITY

Statement of problem

Teachers in many context areas (i.e. social studies, art, English) do not believe, and, in some cases, do not want to believe the computer can be a useful aid in their teaching. They feel that other instructional devices are more manipulable, less expensive, less demanding of teacher skill, and more responsive to their pedagogical needs.

Solution of Problem

Pre-service and inservice training which overcomes these hurdles to widespread instructional use of computers in all subject areas has some, or all, of the following characteristics:

skilled demonstration of use of software appropriate to the objectives of each subject area.

demonstration with a teacher's own class or, at least, with children from the teacher's own grade level or subject area.

demonstration using software that stimulates higher cognitive skills of problem-solving and creative synthesis.

demonstration stressing the unique interactive attributes of drill and practice on the computer, which make even that low-level cognitive stimulus on the computer superior to drill using workbooks or worksheets.

demonstrations of public domain software, inexpensive software, software adaptable to all subject areas (such as word searches and crossword puzzles, game show, etc.), which enable a school to get maximum value for money invested in computer materials because they have cross-discipline potential.

demonstrations of sources of printed programs in various subject areas that may be easily typed to disk by one who knows nothing about programming.

demonstrations of program-authoring "shell" program generators, which a teacher may customize with materials from a current classroom lesson.

demonstrations of class management systems on disks.

demonstrations of computer programs for administrators, counselors, the school nurse, and other school personnel.

display of print materials on computers for teachers, administrators, and students.

demonstrations of exciting peripherals that are attractive to children - the Koala Pad and its software, the Power Pad drawing board, the Mockingboard and its Music Construction Set, other music generating materials, the supersprite board with ampersprites and the Echo II speech synthesizer, with its possibilities for use in special education.

These demonstrations have proved irresistibly attractive to the hundreds of teachers in my workshops over the past three years and to the hundreds of children that come to my Saturday computer classes and computer summer school.

COMPUTER-AIDED TEACHER EVALUATION:

FLANDERS' INTERACTION ANALYSIS ON THE APPLE

JOANNE MARTIN-REYNOLDS, CHARLOTTE SCHERER

BOWLING GREEN STATE UNIVERSITY

By most accounts the best known technique for observing verbal classroom behaviors is the Flanders interaction analysis system (Acheson & Gall, 1980). This technique has been widely used in teacher training programs and inservice programs for the past two decades (Campbell & Barnes, 1969; Rosenshine, 1970, 1976; Flanders, 1976; Acheson & Gall, 1980). Even though the system needs careful explanation to be used with some accuracy with preservice teachers and teachers, it provides the opportunity to compare verbal actions with original intent. Such feedback has been found to be extremely helpful in discussing problem areas with preservice teachers and their cooperating teachers in the field experience programs at Bowling Green State University. However, construction and interpretation of the matrix is time-consuming and sometimes frustrating for the beginner, so much so that a thorough analysis of the data gathered is not usually achieved. This study was undertaken in order to determine whether students might be significantly aided in their evaluation efforts if the building of the matrix and the percentage analyses were done by computer, with student time then devoted to the interpretation of the data, aided by a computer tutorial program.

Since student teachers at BGSU regularly are taught to use Flanders' Interaction Analysis to analyze and evaluate lessons, it was speculated that the computer-calculated Flanders matrix and the tutorial would help them interpret matrix results; student teachers would do more in-depth analyses of their lessons than if they had calculated by hand. After being taught the Flanders system, student teachers were asked to tally a lesson, complete by hand a matrix and calculate the various ratios. They then analyzed the lesson in terms of meeting their objectives. One week later, the student teachers were introduced to the computer-aided Flanders matrix using the computer-aided Flanders program to calculate ratios and interpret results. Once again they analyzed the lesson in terms of meeting their objectives. The two lesson analyses were compared for each student teacher, using the following criteria: 1) number of references to 12 selected items such as teacher talk, student talk, questioning, praise, and extension of student ideas; 2) degree of accuracy of interpretation of data; 3) degree of awareness of types of verbal classroom interaction; 4) number of comments related to objectives; 5) number of comments related to students; 6) number of comments related to self. Students were also given a short opinionnaire on which to express their reactions to the computer program.

Initial results indicated that with the computer-aided Flanders, students tended to make more accurate written analyses of the matrix data in terms of I/D ration, motivation and control, praise, etc. Increases were noted in number of comments related to objectives and students. Mixed results were found for student teacher reactions to the computer program. Ratings of the computer program indicated they found it easier to calculate and understand the data, but they tended not to see a need for such an extended analysis and interpretation.

It was concluded that the program did seem to focus the students' atten-

tion more directly on specific items of analysis and that they generally felt the program to be helpful. Students tended to react negatively to the increased amount of time necessary for the more detailed computer-assisted analysis, but they were still able to see a clear relationship between the Flanders analysis and teaching. Further investigations are in progress to help clarify the extent to which this program increases student ability to accurately assess teaching strategies in terms of objectives.

CORPORATIONS, COMPUTERS, COMMUNITY AND SUMMER TECH '83

ROBERT E. McNEMAR

COLUMBUS PUBLIC SCHOOLS

The nation's largest community computer education program, SUMMER TECH '83, took place July 18 through August 12 at the Columbus Public Schools' Mohawk Middle School located in downtown Columbus, Ohio. The community computer education program was jointly sponsored by the Columbus Public Schools and College of Education of The Ohio State University. Enrolment for the program exceeded the project's original target of 10,000 registrations. There were over fifty different courses in a wide range of computer skills, from Computer for Kids to Assembly Language.

Start-up costs totaling over ½ million dollars were funded in part by local businesses. Because of this and similar kinds of support, the cost of the ten hours of instruction for participants was only twenty dollars.

The Ohio State University, College of Education played a major role in assisting in the development of the curriculum, training teachers and conducting an evaluation of the program. The Columbus Public Schools administered the program and utilized instructors from Columbus Public Schools, surrounding school systems, The Ohio State University and local corporations.

The summer program was open to students of all ages. In classes, elementary school children worked side by side with senior citizens. The youngest SUMMER TECH student was three and the oldest eighty-three.

Plans are being formulated for an enhanced SUMMER TECH '84.

CAI STUDY PROGRAM FOR THE GENERAL PSYCHOLOGY COLLEGE COURSE

TERRY F. PETTIJOHN

OHIO STATE UNIVERSITY

One of the challenging problems of the large-lecture format classes is the inability to provide adequate attention and tutoring of individual students. This project is the development of a CAI help program for the general psychology course. Each unit of the course was available on a disk for students to study at their leisure. Each unit consisted of: a) a list of learning objectives, b) a programmed review session (fill-in the blank), c) learning activities, and d) practice multiple-choice quizzes. The programs were written in MicroPi's PILOT for the IBM PC. Autumn Quarter this study help program was optional for my class of 115 students. I had two IBM PCs available for students to use approximtely 60 hours per week. The students used the study program a mean of 1.7 times per week, and studied about 31.6 minutes per session. A beneficial side effect was that students studied in small groups (mean 2.3 students), and developed friendships. 94% believed that it was a fun experience. Test scores of the class were higher than another section of the course that was not using the computer. Thus, CAI can provide an excellent system of individualized study help for the general psychology college course.

MONITORED SIMULATIONS OF THE CLINICAL ENCOUNTER

JANE SANTRY PIERCE, MICHAEL ALTMAN, SUZANNE BASSANI, BARBARA SKILLMAN

THE OHIO STATE UNIVERSITY COLLEGE OF MEDICINE

Since 1974, the "Computer-Aided Simulation of the Clinical Encounter" (case) program has been available at The Ohio State University College of Medicine. "case" has been used by both students and faculty as a self-assessing competency tool. Based on a free language questioning style, the user assumes the role of physician while the computer is a "patient". The user has the option to question the "patient" at length, perform a physical examination and request laboratory data. At the conclusion of the "case", the user's performance is evaluated in several areas of competency (e.g., proper diagnosis and treatment, critical concept obtainment, and treatment costs).

Faculty members involved with the College of Medicine's Independent Study Program (ISP) felt "case" would be an appropriate learning tool for second-year students used in conjunction with their already existing computerized tutorial modules. It was felt, however, that a more tutorial "case" interaction with a very limited data base focusing on a single clinical problem was necessary at the second-year pre-clinical level. This idea developed into three modular simulations (MODSIMS) in which the student's interaction is closely monitored and, using various educational strategies, the student is guided by a computerized tutor. This tutor is programmed to interrupt and suggest paths of patient questioning, request and analyze defferential diagnoses, and/or aid in problem-solving. The extent and the manner in which the tutor interacts with the student depends solely upon the ISP faculty member who is authoring the MODSIM. At the conclusion of a MODSIM, there is no performance assessment. Each MODSIM ends with a tutorial review of the material covered by the simulation.

Through the ADCIS poster session, we would like to present this new format in medical education. Using four posters, the well-seasoned "case" model will be compared with the newer MODSIM model, outlining major differences in (1) the underlying objectives and (2) the actual interaction between the student and the computer. Via a microcomputer to mainframe connection, an IBM Personal Computer will be available during the session allowing demonstrations of both programs.

A CAI SYSTEM FOR SIGHT-SINGING

MARTIN PRÉVEL

UNIVERSITÉ LAVAL, QUÉBEC

Sight singing along with musical dictation, constitutes the basis of a professional musician's training. The addition, to musical training systems, of a device capable of detecting notes sung by the human voice permits the integration into CAI of an important aspect of musical training which heretofore has been neglected because of a lack of necessary hardware. The MELOCAPTEUR is a device we have developed to fill this gap.

As our primary objective is restricted to the detection of melodic degrees emited by the human voice with a relatively limited register and at a moderate tempo, it appears possible and opportune to develope an original device whose cost would be much less than the cost of the microcomputer with which it would be used. The subject of this poster session is a presentation of the pedagogic and technical aspects of the pitch extractor peripheral we have developed within our CAL research program in music.

The ability to digitize musical intonation implies, above all else, the capacity to detect the fundamentals of sounds which may be more or less rich in overtones. The few pitch detector systems which exist at present were developed in relation to scientific applications in which it is important to extract all aspects of a sound signal, nc matter what the sound source is. The complexity of these systems renders them inaccessible for the field of computer assisted instruction in music because of their high cost.

The MELOCAPTEUR was developed originally as a peripheral for EXERCETTE, a microcomputer we have developed exclusively for ear training. An adaptation of the MELOCAPTEUR has been developed to function on the APPLE II computer and a third version is planned for the IBM PC and those computers which are compatible with it.

S T A R S (SPACE TIME ARMY RECONNAISSANCE SYSTEM)

THOMAS C. REEVES, UNIVERSITY OF GEORGIA

WILLIAM D. AGGEN, UNIVERSITY OF MARYLAND

The Center for Instructional Development and Evaluation at the University of Maryland University College has a contract with the U.S. Army Continuing Education System in Europe to develop and implement a military-specific microcomputer/videodisc integrated instructional program in functional literacy. The Computer-Based Instruction Project, on-going since 1979, has for the 1983-84 contract year two major components:

1. The implementation and evaluation of computer-based functional literacy lessons at ten Army sites in the USAREUR command. These lessons, designed specifically for wheel vehicle mechanics, armor crewmen, and missle crewmen are the subject of an extensive evaluation designed to document the system usage, collect information for lesson improvement, assess the immediate learner effectiveness of the lessons, and judge the long range impact of the instructional programs.

2. The design, development, demonstration, and evaluation of improved microcomputer/videodisc instructional programs in functional literacy. This component includes the following:

A. a hardware configuration which includes single screen integration of videodisc images and high resolution computer-generated material, digitized audio, touch sensitive screen, and a hard disk storage unit (DEC IVIS);

B. two time-travel simulations in which soldiers journey through time to exciting environments, e.g., the Battle of the Bulge (STARS I) and a futuristic conflict (STARS II) where they must apply functional literacy skills to accomplish their missions;

C. state-of-the-art functional literacy instruction which emphasizes learning basic skills from a problem solving perspective in relevant military contexts;

D. unique videodisc production strategies including extensive use of historical footage, multiple channel audio, state-of-the-art special effects, and new image-to-videotape transfer techniques using computer controlled video animation systems.

The second component also involves an extensive evaluation effort. One innovative aspect of this evaluation is that, instead of using the multiple-choice pretests and posttests found in many CBI programs, these lessons employ problems in simulated environments for diagnosis of student needs and assessment of immediate effectiveness. For example, students are called upon to calculate the amount of "Time Warp Energy Units" they will need to proceed with their mission using basic multiplication skills. If they correctly calculate the amount, the students proceed with their mission. If not, they are immediately presented with instruction matched to the required functional math skill.

USING MICROCOMPUTERS TO TEACH INTRODUCTORY STATISTICS

SAM SHERRILL

UNIVERSITY OF CINCINNATI

PROJECT

Introductory descriptive and inductive statistics for planners was taught using business computers (Model III's) manufactured by the Tandy Corporation. A local Radio Shack Computer Center made both machines and a classroom available one day per week (without charge) for three hours during each of the ten and a half week quarters. Each student received about one and a half hours of standard classroom instruction and one and a half hours microcomputer instruction at the Radio Shack Computer Center. None of the thirty-seven students had prior experience with microcomputers and the large majority had only very limited prior instruction in mathematics. They were told that the main purpose of the class was to introduce them to the statistical use of microcomputers.

Students were taught some elementary BASIC language commands and subroutines at the beginning of the course (mainly input statements, loops, and arrays). They were required to use this elementary knowledge and their own resident ability to reason logically to write small statistical programs. One of the purposes of these exercises was to expose them to the interior of statistical formulas and calculations in a meaningful way. The traditional derivation of statistical formulae seems to be a meaningless exercise (designed covertly perhaps to torture students) that is almost immediately forgotten by nearly all those exposed to it. Writing a program to do the necessary arithmetic calculations requires the student to closely examine the formula and reproduce it in the different medium of a program. Efficient programming occasionally requires an algebraic modification of the formula; thus, the student not only learns the formula but learns to modify it for a meaningful purpose.

Specific results will be presented at this session. Material used in class and a microcomputer will also be available for review and discussion.

AUTHOR TRANSPORTABILITY: ADVANTAGES AND DISADVANTAGES

TRACIE M. JENKINS, MACON JUNIOR COLLEGE

SYLVIA SILVER, GEORGE WASHINGTON UNIVERSITY MEDICAL CENTER

Although the content of computer-assisted instruction may be universal, the implementation is generally unique to a specific computer system. The difficulty arises from the nontransportability of software between different computers and solutions to the problem are slow in coming. An alternative concept is to transport the author rather than the software. This permits an adjustment of the instructional product to differences in capabilities and technology between one computer system and another. Transporting an author to a new environment may also introduce some additional insights into computer-assisted instruction. This paper will discuss the philosophy of author transportability and mechanisms for achieving this goal.

INTERACTIVE VIDEODISC

TIMOTHY W. SPANNAUS

CREATIVE UNIVERSAL, INC.

Creative Universal, Inc., working with the Sony Corporation, has developed a highly interactive videodisc training system.

The system hardware incorporates videodisc, microcomputer, digitized audio adapter and light pen. Software includes an extended version of Pilot running under CP/M and custom courseware. Unique features of the system are that it requires no typing on the part of the learner and that it reduces the role of reading in the instruction.

The courseware incorporates training segments, still checks, simulations, and testing in a highly interactive self-instructional environment. The structure of the courseware allows use of the program for training or reference, depending on whether the system or the learner controls the path through the course.

Field tests of prototypes indicate high levels of learner acceptance and performance improvement.

DEVELOPING SUPPORT MATERIALS FOR GREATER LEARNER CONTROL OF COMPUTER-BASED INSTRUCTION

BRENT G. WILSON

NORTHERN ILLINOIS UNIVERSITY

DESCRIPTION

In order for learners to be able to use computer-based instructional materials in a truly individualized way, they must be able to adapt it to their needs; that is, they must be able to exert some degree of control over the events of instruction. Learner control over computer-based instruction is inhibited, however, by two common limitations: (1) limited ability to randomly access information displays presented to learners, and (2) limited ability to display large amounts of information at any given time. To help increase the learner's control over instruction, two suggestions are proposed: (1) provide denser information display through the use of diagrams and graphics that summarize concepts in a subject matter, and (2) provide more support materials in hard-copy form such as job aids, reference cards, and content summaries to allow easy access for review and reference purposes. To demonstrate the value of these recommendations, a computer-assisted lesson introducing learners to kinds of electrical circuits has been developed. Support materials include a tree diagram of concepts in the lesson. The strategies and rationale behind the product will be discussed in the poster session, as well as evaluative data supporting the usefulness of the materials. The computer-based lesson will later be used as part of a larger-scale research project.

I will be using an Apple IIe computer system to demonstrate the program.

SOFTWARE DEVELOPMENT FOR THE HOME EC CLASSROOM

CLO WILTSE

MONTANA STATE UNIVERSITY

"Computers in Home Economics" was the title of one-week workshops sponsored by the Home Economics Department of the Montana State University in June on the campus at Bozeman and in Glassow, Montana, in July of 1983.

Taught by Clo Wiltse (speaker), a high school Home Economics/Computer Science teacher, it consisted of 30 members, from college students to classroom teachers and extension agents with considerable years of experience.

Students were introduced to the computer, its place in our lives, and implications for the families and homes of the future.

Over 1200 dollars-worth of software was purchased with a grant from the Montana Teaching Center (now defunct), and members of the class were charged with reviewing and evaluating these materials.

From their experiences in running the collection of software available, they were able to contribute ably to discussions on the criteria needed in software for the classroom.

Each member was encouraged to write her own program for use in the home economics classroom. At the conclusion of the week, each presented her own completed "running" program within the context of the lesson unit for which it was written. Role playing was used with the rest of the class being the "students" and operating the program when assigned by their "teacher".

The applications chosen for the use of a computer, the variety of uses, and the cleverness of many of the programs developed in this one-week's time was most promising and stimulative.

Due to the variety and quality of the computer applications produced, the instructor invited many of the members to collaborate in the production of the "Classroom Potpourri" software program which she markets for them under her "clo's - - - - -line" software company. The compilation is receiving wide acceptance, as it is a practical way for the classroom teacher to introduce the use of computers into her teaching techniques.